```
LABEL  OPROGRAMOLISTING00177202CC USER=SITE    ; COMPILE FORTRAN/DISK WITH ALGOL LIBRARY:ALGOL STACK=100 ALGOL  /FORTRAN
                BURROUGHS B-5700 ALGOL COMPILER MARK XVI.0.00   THURSDAY, 07/21/77,  11:03 AM.


                                FORTRAN/DISK
                                ============


% F O R T R A N  C O M P I L E R   XVI.0.00                  10/01/74          00000000 T  0000
  COMMENT: * TITLE: B5500/B5700 MARK XVI SYSTEM RELEASE                  *   00001000 T  0000
           * FILE ID: SYMBOL/FORTRAN TAPE ID: SYMBOL1/FILE000            *   00001010 T  0000
           * THIS MATERIAL IS PROPRIETARY TO BURROUGHS CORPORATION       *   00001011 T  0000
           * AND IS NOT TO BE REPRODUCED, USED, OR DISCLOSED             *   00001012 T  0000
           * EXCEPT IN ACCORDANCE WITH PROGRAM LICENSE OR UPON           *   00001013 T  0000
           * WRITTEN AUTHORIZATION OF THE PATENT DIVISION OF             *   00001014 T  0000
           * BURROUGHS CORPORATION, DETROIT, MICHIGAN 48232              *   00001015 T  0000
           *                                                             *   00001016 T  0000
           * COPYRIGHT (C)  1971, 1972, 1974                             *   00001017 T  0000
           *       BURROUGHS CORPORATION                                 *   00001018 T  0000
           *  AA320206   AA393180   AA332366                             *;  00001019 T  0000
% BEWARE ALL YE WHO SEEK HEREIN:                                         %997-00001020 T  0000
% THIS COMPILER IS A MESS.  THE CONCEPTS OF CHARACTER CONVERSION FROM          00001100 C  0000
% HOLLERITH (= EBCDIC), SCANNING, AND PARSING ARE NOT CLEARLY SEPARATED       00001105 C  0000
% IN THE CODE.  FOR EXAMPLE, PROCEDURE "SCAN" TRIES TO DO ALL 3 AT ONCE.      00001110 C  0000
% DAN ROSS    UCSC    APRIL 12, 1977                                  %997-00001115 C  0000
COMMENT                                                                      00001120 C  0000
                                                                             00001900 T  0000
                        FORTRAN ERROR MESSAGES                               00002000 T  0000

000   SYNTAX ERROR                                                           00003000 T  0000
001   MISSING OPERATOR OR PUNCTUATION                                        00004000 T  0000
002   CONFLICTING COMMON AND/OR EQUIVALENCE ALLOCATION                       00005000 T  0000
003   MISSING RIGHT PARENTHESIS                                              00006000 T  0000
004   ENTRY STMT ILLEGAL IN MAIN PGM OR BLOCK DATA                           00007000 T  0000
005   MISSING END STATEMENT                                                  00008000 T  0000
006   ARITHMETIC EXPRESSION REQUIRED                                         00009000 T  0000
007   LOGICAL EXPRESSION REQUIRED                                            00010000 T  0000
008   TOO MANY LEFT PARENTHESES                                              00011000 T  0000
009   TOO MANY RIGHT PARENTHESES                                             00012000 T  0000
010   FORMAL PARAMETER ILLEGAL IN COMMON                                     00013000 T  0000
011   FORMAL PARAMETER ILLEGAL IN EQUIVALENCE                                00014000 T  0000
012   THIS STATEMENT ILLEGAL IN BLOCK DATA SUBPROGRAM                        00015000 T  0000
013   INFO ARRAY OVERFLOW                                                    00016000 T  0000
014   IMPROPER DO NEST                                                       00017000 T  0000
015   DO LABEL PREVIOUSLY DEFINED                                            00018000 T  0000
016   UNRECOGNIZED STATEMENT TYPE                                            00019000 T  0000
017   ILLEGAL DO STATEMENT                                                   00020000 T  0000
018   FORMAT STATEMENT MUST HAVE LABEL                                       00021000 T  0000
019   UNDEFINED LABEL                                                        00022000 T  0000
020   MULTIPLE DEFINITION                                                    00023000 T  0000
021   ILLEGAL IDENTIFIER CLASS IN THIS CONTEXT                               00024000 T  0000
022   UNPAIRED QUOTES IN FORMAT                                              00025000 T  0000
023   NOT ENOUGH SUBSCRIPTS                                                  00026000 T  0000
024   TOO MANY SUBSCRIPTS                                                    00027000 T  0000
025   FUNCTION OR SUBROUTINE PREVIOUSLY DEFINED                              00028000 T  0000
026   FORMAL PARAMETER MULTIPLY DEFINED IN HEADING                           00029000 T  0000
```

```
027   ILLEGAL USE OF NAMELIST                                    00030000 T  0000
028   NUMBER OF PARAMETERS INCONSISTENT                          00031000 T  0000
029   CANNOT BRANCH TO FORMAT STATEMENT                          00032000 T  0000
030   SUBROUTINE OR FUNCTION NOT DEFINED IN PROGRAM              00033000 T  0000
031   IDENTIFIER ALREADY GIVEN TYPE                              00034000 T  0000
032   ILLEGAL FORMAT SYNTAX                                      00035000 T  0000
033   INCORRECT USE OF FILE                                      00036000 T  0000
034   INCONSISTENT USE OF IDENTIFIER                             00037000 T  0000
035   ARRAY IDENTIFIER EXPECTED                                  00038000 T  0000
036   EXPRESSION VALUE REQUIRED                                  00039000 T  0000
037   ILLEGAL FILE CARD SYNTAX                                   00040000 T  0000
038   ILLEGAL CONTROL ELEMENT                                    00041000 T  0000
039   DECLARATION MUST PRECEDE FIRST REFERENCE                   00042000 T  0000
040   INCONSISTENT USE OF LABEL AS PARAMETER                     00043000 T  0000
041   NO. OF PARAMS. DISAGREES WITH PREV. REFERENCE              00044000 T  0000
042   ILLEGAL USE OF FORMAL PARAMETER                            00045000 T  0000
043   ERROR IN HOLLERITH LITERAL CHARACTER COUNT                 00046000 T  0000
044   ILLEGAL ACTUAL PARAMETER                                   00047000 T  0000
045   TOO MANY SEGMENTS IN SOURCE PROGRAM                        00048000 T  0000
046   TOO MANY PRT ASSIGNMENTS IN SOURCE PROGRAM                 00049000 T  0000
047   LAST BLOCK DECLARATION HAD LESS THAN 1024 WORDS            00050000 T  0000
048   ILLEGAL I/O LIST ELEMENT                                   00051000 T  0000
049   LEFT SIDE MUST BE SIMPLE OR SUBSCRIPTED VARIABLE           00052000 T  0000
050   VARIABLE EXPECTED                                          00053000 T  0000
051   ILLEGAL USE OF .OR.                                        00054000 T  0000
052   ILLEGAL USE OF .AND.                                       00055000 T  0000
053   ILLEGAL USE OF .NOT.                                       00056000 T  0000
054   ILLEGAL USE OF RELATIONAL OPERATOR                         00057000 T  0000
055   ILLEGAL MIXED TYPES                                        00058000 T  0000
056   ILLEGAL EXPRESSION STRUCTURE                               00059000 T  0000
057   ILLEGAL PARAMETER                                          00060000 T  0000
058   RECORD BLOCK GREATER THAN 1023                             00061000 T  0000
059   TOO MANY OPTIONAL FILES                                    00062000 T  0000
060   FILE CARDS MUST PRECEDE SOURCE DECK                        00063000 T  0000
061   BINARY WRITE STATEMENT HAS NO LIST                         00064000 T  0000
062   UNDEFINED FORMAT NUMBER                                    00065000 T  0000
063   ILLEGAL EXPONENT IN CONSTANT                               00066000 T  0000
064   ILLEGAL CONSTANT IN DATA STATEMENT                         00067000 T  0000
065   MAIN PROGRAM MISSING                                       00068000 T  0000
066   PARAMETER MUST BE ARRAY IDENTIFIER                         00069000 T  0000
067   PARAMETER MUST BE EXPRESSION                               00070000 T  0000
068   PARAMETER MUST BE LABEL                                    00071000 T  0000
069   PARAMETER MUST BE FUNCTION IDENTIFIER                      00072000 T  0000
070   PARAMETER MUST BE FUNCTION OR SUBROUTINE ID                00073000 T  0000
071   PARAMETER MUST BE SUBROUTINE IDENTIFIER                    00074000 T  0000
072   PARAMETER MUST BE ARRAY IDENTIFIER OR EXPRESSION           00075000 T  0000
073   ARITHMETIC - LOGICAL CONFLICT ON STORE                     00076000 T  0000
074   ARRAYID MUST BE SUBSCRIPTED IN THIS CONTEXT                00077000 T  0000
075   MORE THAN ONE MAIN PROGRAM                                 00078000 T  0000
076   ONLY COMMON ELEMENTS PERMITTED                             00079000 T  0000
077   TOO MANY FILES                                             00080000 T  0000
078   FORMAT OR NAMELIST TOO LONG                                00081000 T  0000
079   FORMAL PARAMETER MUST BE ARRAY IDENTIFIER                  00082000 T  0000
080   FORMAL PARAMETER MUST BE SIMPLE VARIABLE                   00083000 T  0000
081   FORMAL PARAMETER MUST BE FUNCTION IDENTIFIER               00084000 T  0000
082   FORMAL PARAMETER MUST BE SUBROUTINE IDENTIFIER             00085000 T  0000
083   FORMAL PARAMETER MUST BE FUNCTION OR SUBROUTINE            00086000 T  0000
```

```
084   DO OR IMPLIED DO INDEX MUST BE INTEGER OR REAL                    00087000 T  0000
085   ILLEGAL COMPLEX CONSTANT                                          00088000 T  0000
086   ILLEGAL MIXED TYPE STORE                                          00089000 T  0000
087   CONSTANT EXCEEDS HARDWARE LIMITS                                  00090000 T  0000
088   PARAMETER TYPE CONFLICTS WITH PREVIOUS USE                        00091000 T  0000
089   COMPLEX EXPRESSION ILLEGAL IN IF STATEMENT                        00092000 T  0000
090   COMPLEX EXPRESSION ILLEGAL IN RELATION                            00093000 T  0000
091   TOO MANY FORMATS REFERENCED BUT NOT YET FOUND                     00094000 T  0000
092   VARIABLE ARRAY BOUND MUST BE FORMAL VARIABLE                      00095000 T  0000
093   ARRAY BOUND MUST HAVE INTEGER OR REAL TYPE                        00096000 T  0000
094   COMMA OR RIGHT PARENTHESIS EXPECTED                               00097000 T  0000
095   ARRAY ALREADY GIVEN BOUNDS                                        00098000 T  0000
096   ONLY FORMAL ARRAYS MAY BE GIVEN VARIABLE BOUNDS                   00099000 T  0000
097   MISSING LEFT PARENTHESIS IN IMPLIED DO                            00100000 T  0000
098   SUBSCRIPT MUST BE INTEGER OR REAL                                 00101000 T  0000
099   ARRAY SIZE CANNOT EXCEED 32767 WORDS                              00102000 T  0000
100   COMMON OR EQUIV BLOCK CANNOT EXCEED 32767 WORDS                   00103000 T  0000
101   THIS STATEMENT ILLEGAL IN LOGICAL IF                              00104000 T  0000
102   REAL OR INTEGER TYPE REQUIRED                                     00105000 T  0000
103   ARRAY BOUND INFORMATION REQUIRED                                  00106000 T  0000
104   REPLACEMENT OPERATOR EXPECTED                                     00107000 T  0000
105   IDENTIFIER EXPECTED                                               00108000 T  0000
106   LEFT PARENTHESIS EXPECTED                                         00109000 T  0000
107   ILLEGAL FORMAL PARAMETER                                          00110000 T  0000
108   RIGHT PARENTHESIS EXPECTED                                        00111000 T  0000
109   STATEMENT NUMBER EXPECTED                                         00112000 T  0000
110   SLASH EXPECTED                                                    00113000 T  0000
111   ENTRY STATEMENT CANNOT START PROGRAM UNIT                         00114000 T  0000
112   ARRAY MUST BE DIMENSIONED PRIOR TO EQUIV STMT                     00115000 T  0000
113   INTEGER CONSTANT EXPECTED                                         00116000 T  0000
114   COMMA EXPECTED                                                    00117000 T  0000
115   SLASH OR END OF STATEMENT EXPECTED                                00118000 T  0000
116   FORMAT, ARRAY OR NAMELIST EXPECTED                                00119000 T  0000
117   END OF STATEMENT EXPECTED                                         00120000 T  0000
118   IO STATEMENT WITH NAMELIST CANNOT HAVE IO LIST                    00121000 T  0000
119   COMMA OR END OF STATEMENT EXPECTED                                00122000 T  0000
120   STRING TOO LONG                                                   00123000 T  0000
121   MISSING QUOTE AT END OF STRING                                    00124000 T  0000
122   ILLEGAL ARRAY BOUND                                               00125000 T  0000
123   TOO MANY HANGING BRANCHES                                         00126000 T  0000
124   TOO MANY COMMON OR EQUIVALENCE ELEMENTS                           00127000 T  0000
125   ASTERISK EXPECTED                                                 00128000 T  0000
126   COMMA OR SLASH EXPECTED                                           00129000 T  0000
127   DATA SET TOO LARGE                                                00130000 T  0000
128   TOO MANY ENTRY STATEMENTS IN THIS SUBPROGRAM                      00131000 T  0000
129   DECIMAL WIDTH EXCEEDS FIELD WIDTH                                 00132000 T  0000
130   UNSPECIFIED FIELD WIDTH                                           00133000 T  0000
131   UNSPECIFIED SCALE FACTOR                                          00134000 T  0000
132   ILLEGAL FORMAT CHARACTER                                          00135000 T  0000
133   UNSPECIFIED DECIMAL FIELD                                         00136000 T  0000
134   DECIMAL FIELD ILLEGAL FOR THIS SPECIFIER                          00137000 T  0000
135   ILLEGAL LABEL                                                     00138000 T  0000
136   UNDEFINED NAMELIST                                                00139000 T  0000
137   MULTIPLY DEFINED ACTION LABELS                                    00140000 T  0000
138   TOO MANY NESTED DO STATEMENTS                                     00141000 T  0000
139   STMT FUNCTION ID AND EXPRESSION DISAGREE IN TYPE                  00142000 T  0000
140   ILLEGAL USE OF STATEMENT FUNCTION                                 00143000 T  0000
```

```
141   UNRECOGNIZED CONSTRUCT                                                    00143001 T  0000
142   RETURN, STOP OR CALL EXIT REQUIRED IN SUBPROGRAM                          00143002 T  0000
143   FORMAT NUMBER USED PREVIOUSLY AS LABEL                                    00143003 T  0000
144   LABEL USED PREVIOUSLY AS FORMAT NUMBER                                    00143004 T  0000
145   NON-STANDARD RETURN REQUIRES LABEL PARAMETERS                             00143005 T  0000
146   DOUBLE OR COMPLEX REQUIRES EVEN OFFSET                                    00143006 T  0000
147   FORMAL PARAMETER ILLEGAL IN DATA STATEMENT                                00143007 T  0000
148   TOO MANY LOCAL VARIABLES IN SOURCE PROGRAM                                00143008 T  0000
149   A $FREEFORM SOURCE LINE MUST HAVE < 67 COLS                               00143009 T  0000
150   A HOL/STRING UNDER $FREEFORM MUST BE ON ONE LINE                          00143010 T  0000
151   THIS CONSTRUCT IS ILLEGAL IN TSS FORTRAN                                  00143011 T  0000
152   ILLEGAL FILE CARD PARAMETER VALUE                                         00143012 T  0000
153   RETURN IN MAIN PROGRAM NOT ALLOWED                                        00143013 T  0000
154   NON-POSITIVE SUBSCRIPTS ARE ILLEGAL                                       00143014 T  0000
155   NON-IDENTIFIER USED FOR NAME OF LIBRARY ROUTINE                           00143015 T  0000
156   HYPHEN EXPECTED                                                           00143016 T  0000
157   SEQUENCE NUMBER EXPECTED                                                  00143017 T  0000
158   TOO MANY RECURSIVE CALLS ON LIBRARY ROUTINES                              00143018 T  0000
159   ASTERISK NOT ALLOWED IN READ STATEMENT                                    00143019 T  0000
160   ASTERISK ONLY ALLOWED IN FREE FIELD OUTPUT                                00143020 T  0000
161   TOO MANY *'ED LIST ELEMENTS IN OUTPUT                                     00143021 T  0000
162   HOL OR QUOTED STRING GREATER THAN 7 CHARACTERS IN EXPRESSION              00143022 T  0000
164   PLUS NOT ALLOWED IN THIS FORMAT PHRASE                                    00143024 T  0000
165   K NOT ALLOWED IN THIS FORMAT PHRASE                                       00143025 T  0000
166   $ NOT ALLOWED IN THIS FORMAT PHRASE                                       00143026 T  0000
167   INTRINSICS-NAMED FUNCT MUST HAVE PREVIOUS EXTERNAL REFERENCE.             00143027 T  0000
168   DATA STMT COMMON ELEM MUST BE IN BLK DATA SUBPRM                          00143028 T  0000
169   VARIABLE CANNOT BE A FORMAL PARAMETER OR IN COMMON                        00143029 T  0000
170   CURRENT SUBPROGRAM ID EXPECTED                                            00143030 T  0000
171   SUBPROGRAM, EXTERNAL, OR ARRAY ID EXPECTED                                00143031 T  0000
172   REPEAT, WIDTH, AND DECIMAL PARTS MUST BE LESS THAN 4091                   00143032 T  0000
173   REPEAT PART MUST BE EMPTY OR >0 AND < 4091                                00143033 T  0000
174   IN SUBPRGM:VARBL IS USED PRIOR TO USE IN DATSTMT                          00143034 T  0000
175   SYNTATICAL TOKEN CONTAINS TOO MANY CHARACTERS.                            00143035 T  0000
176   INTEGER VALUE OF 8 EXPECTED                                               00143036 T  0000
177   INTEGER VALUE OF 4 EXPECTED                                               00143037 T  0000
178   INTEGER VALUE OF 4 OR 8 EXPECTED                                          00143038 T  0000
179   AN ALPHABETIC LETTER (A,B,C,...,Z) IS REQUIRED                            00143039 T  0000
180   SECOND RANGE LETTER MUST BE GREATER THAN FIRST                            00143040 T  0000
181   IMPLICIT MUST BE FIRST STATEMENT IN PROGRAM UNIT                          00143041 T  0000
182   REAL/INTEGER/LOGICAL/COMPLEX/DOUBLEPRECISION REQ                          00143042 T  0000
183   ILLEGAL USE OF ASTERISK FOR RUN-TIME EDITING                      %111-   00143043 C  0000
184   NO PROGRAM UNIT FOR THIS END STATEMENT                            %112-   00143044 C  0000
      ;                                                                         00143999 T  0000
      BEGIN                                                                     00144000 T  0000
PRT(22) = *LIST, LABEL, OR SEGMENT DESCRIPTOR*
PRT(23) = *OUTER BLOCK DESCRIPTOR*
PRT(24) = *SEGMENT DESCRIPTOR*

                                                          START OF SEGMENT **********    2
      INTEGER ERRORCT; COMMENT MUST BE FIRST DECLARED VARIABLE;                 00145000 T  0000
PRT(25) = ERRORCT
      INTEGER SAVETIME; COMMENT MUST BE 2 ND DECLARED VARIABLE;                 00146000 T  0000
PRT(26) = SAVETIME
      INTEGER CARDCOUNT,ESTIMATE,SEQERRCT ;                                     00147000 T  0000
PRT(27) = CARDCOUNT
PRT(30) = ESTIMATE
PRT(31) = SEQERRCT
```

```
                    INTEGER SWARYCT;                                                    00147500 T  0000
PRT(32) = SWARYCT
                    REAL TWODPRTX;                                                      00148000 T  0000
PRT(33) = TWODPRTX
                    REAL INITIALSEGNO;                                                  00149000 T  0000
PRT(34) = INITIALSEGNO
                    REAL NEXT, FNEXT, NAME, NUMTYPE;                                    00150000 T  0000
PRT(35) = NEXT
PRT(36) = FNEXT
PRT(37) = NAME
PRT(40) = NUMTYPE
                    REAL A, B, I, J, P, T, TS, Z;                                       00151000 T  0000
PRT(41) = A
PRT(42) = B
PRT(43) = I
PRT(44) = J
PRT(45) = P
PRT(46) = T
PRT(47) = TS
PRT(50) = Z
                    REAL RTI;           % START COMPILE TIME                            00152000 T  0000
PRT(51) = RTI
                    REAL EXPVALUE, EXPRESULT, EXPLINK;                                  00153000 T  0000
PRT(52) = EXPVALUE
PRT(53) = EXPRESULT
PRT(54) = EXPLINK
                    REAL SPLINK, FUNVAR;                                                00155000 T  0000
PRT(55) = SPLINK
PRT(56) = FUNVAR
                    REAL DATAPRT,DATASTRT,DATALINK,DATASKP;                             00155100 T  0000
PRT(57) = DATAPRT
PRT(60) = DATASTRT
PRT(61) = DATALINK
PRT(62) = DATASKP
                    BOOLEAN SCANENTER ; % TRUE IFF SCAN JUST DID AN ENTER ON AN ID.     00155110 T  0000
PRT(63) = SCANENTER
                    DEFINE NUMINTM1=                                                    00155210 T  0000
                    83                                                                  00155211 T  0000
                    #;% NUMINTM1 IS THE NUMBER OF INTRINSICS MINUS 1; FOR EACH NEW INTRINSIC  00155212 T  0000
                    %   WHICH IS ADDED, ADD 1 TO THE VALUE ON SEQ# 00155211.            00155213 T  0000
                    DEFINE MAXEL=15#;                                                   00156000 T  0000
                    REAL ARRAY ENTRYLINK[0:MAXEL];                                      00157000 T  0000
PRT(64) = ENTRYLINK
                    REAL ELX;                                                           00158000 T  0001
PRT(65) = ELX
                    ALPHA FNEW, NNEW;                                                   00159000 T  0001
PRT(66) = FNEW
PRT(67) = NNEW
                    REAL LABELMOM;                                                      00160000 T  0001
PRT(70) = LABELMOM
                    INTEGER LOCALS,PARMS,PRTS,FLAGROUTINECOUNTER ;                      00161000 T  0001
PRT(71) = LOCALS
PRT(72) = PARMS
PRT(73) = PRTS
PRT(74) = FLAGROUTINECOUNTER
                    REAL LIMIT, VOIDTSEQ;                                               00161010 T  0001
PRT(75) = LIMIT
```

```
PRT(76) = VOIDTSEQ
                REAL IDINFO, TYPE, NSEG;                                        00162000 T  0001
PRT(77) = IDINFO
PRT(100) = TYPE
PRT(101) = NSEG
                REAL FX1, FX2, FX3, NX1, NX2, NX3;                              00163000 T  0001
PRT(102) = FX1
PRT(103) = FX2
PRT(104) = FX3
PRT(105) = NX1
PRT(106) = NX2
PRT(107) = NX3
                INTEGER LENGTH, LOWERBOUND, GROUPPRT;                           00164000 T  0001
PRT(110) = LENGTH
PRT(111) = LOWERBOUND
PRT(112) = GROUPPRT
                ALPHA EQVID, INTID, REALID ;                                    00165000 T  0001
PRT(113) = EQVID
PRT(114) = INTID
PRT(115) = REALID
                INTEGER ROOT, ADR;                                              00166000 T  0001
PRT(116) = ROOT
PRT(117) = ADR
                INTEGER LASTMODE ;  %%% = 1 FOR $CARD; = 2 FOR $TAPE.           00166100 T  0001
PRT(120) = LASTMODE
                BOOLEAN ERRORTOG,%PREVIOUS LISTED RECORD.                       00167000 T  0001
PRT(121) = ERRORTOG
                        EOSTOG,%END OF STMT TOGGLE.                             00168000 T  0001
PRT(122) = EOSTOG
                        CODETOG,%LIST CODE GENERATED..                          00168200 T  0001
PRT(123) = CODETOG
                        TAPETOG,%MERGE TAPE INPUT.                              00168400 T  0001
PRT(124) = TAPETOG
                        EODS,%END OF DECLRSTMTS, RESET BY ENDS.                 00168500 T  0001
PRT(125) = EODS
                        FILETOG,%PROCESSING FILE CARDS.                         00168600 T  0001
PRT(126) = FILETOG
                        DEBUGTOG,%TO LIST OUT ENTERING AND EXITING PROCEDURES.  00169000 T  0001
PRT(127) = DEBUGTOG
                        DESCREQ,%DESCRIPTOR REQUIRED.                           00169020 T  0001
PRT(130) = DESCREQ
                        XREF,%TO CREATE XREF OF PROGRAM.                        00169030 T  0001
PRT(131) = XREF
                        SAVETOG;%VARIOUS BOOLEANS.                              00169050 T  0001
PRT(132) = SAVETOG
            DEFINE  LIBTAPE      = SAVETOG.[47:1]#,%USE TO SAVE NEW WHILE INCLUDE.          00169053 T  0001
                    SAVEDEBUGTOG = SAVETOG.[46:1]#,%SAVE DEBUGTOG WHILE IN $INCLUDE.       00169054 T  0001
                    SAVECODETOG  = SAVETOG.[45:1]#,%SAVE CODETOG WHILE IN $INCLUDE.        00169055 T  0001
                    SAVEPRTTOG   = SAVETOG.[44:1]#,%SAVE PRTOG WHILE IN $INCLUDE.          00169056 T  0001
                    SAVELISTOG   = SAVETOG.[43:1]#,%SAVE LISTOG WHILE IN $INCLUDE.         00169057 T  0001
                    VOIDTOG      = SAVETOG.[42:1]#,%TO VOID CARDS OR TAPE                  00169058 T  0001
                    DATASTMTFLAG = SAVETOG.[41:1]#,%WHILE IN DATA STMTS.                   00169059 T  0001
                    F2TOG        = SAVETOG.[40:1]#,%ADDR REL TO F+2.                       00169060 T  0001
                    CHECKTOG     = SAVETOG.[39:1]#,%SEQ # CHECK.                           00169061 T  0001
                    LISTOG       = SAVETOG.[38:1]#,%TO LIST OUTPUT.                        00169062 T  0001
                    LISTLIBTOG   = SAVETOG.[37:1]#,%TO LIST LIBRARY OUTPUT.                00169063 T  0001
                    SEQTOG       = SAVETOG.[36:1]#,%TO RESEQ INPUTS.                       00169064 T  0001
```

```
SEQERRORS     = SAVETOG.[35:1]#,%IF SEQUENCE ERRORS.                          00169065 T  0001
TIMETOG       = SAVETOG.[34:1]#,%TO LIST WRAPUP INFO ONLY.                    00169066 T  0001
PRTOG         = SAVETOG.[33:1]#,%TO LIST STORAGE MAP.                         00169067 T  0001
NEWTPTOG      = SAVETOG.[32:1]#,%CREATE NEW SYMBOLIC TAPE.                    00169068 T  0001
SINGLETOG     = SAVETOG.[31:1]#,%TO LIST OUTPUT SINGLE SPACED.                00169069 T  0001
SEGOVFLAG     = SAVETOG.[30:1]#,%SEGMENT OVERFLOW.                            00169070 T  0001
FIRSTCALL     = SAVETOG.[29:1]#,%TO LIST COMPILER HEADING.                    00169071 T  0001
RETURNFOUND   = SAVETOG.[28:1]#,%RETURN,CALL EXIT,STOP FOUND.                 00169072 T  0001
ENDSEGTOG     = SAVETOG.[27:1]#,%END OF SEGMENT.                              00169073 T  0001
LOGIFTOG      = SAVETOG.[26:1]#,%PROCESSING LOGICAL IF STMT.                  00169074 T  0001
NOTOPIO       = SAVETOG.[25:1]#,                                              00169075 T  0001
DATATOG       = SAVETOG.[24:1]#,%WHILE IN DATA STMTS.                         00169076 T  0001
FREEFTOG      = SAVETOG.[23:1]#,%FREE FIELD INPUT.                            00169077 T  0001
SEGPTOG       = SAVETOG.[22:1]#,%DONT PAGE AFTER SUBPRGM                      00169078 T  0001
GLOBALNAME    = SAVETOG.[21:1]#,%TO WRITE ALL IDENTIFIERS.                    00169079 T  0001
NAMEDESC      = SAVETOG.[20:1]#,%IF GLOBALNAME OR LOCALNAME TRUE.             00169080 T  0001
LOCALNAME     = SAVETOG.[19:1]#,%TO WRITE AN IDENTIFIER.                      00169081 T  0001
TSSEDITOG     = SAVETOG.[18:1]#,%TSSEDIT.                                     00169082 T  0001
UNPRINTED     = SAVETOG.[17:1]#,%TO LIST CARD,FALSE IF LISTED.                00169083 T  0001
DCINPUT       = SAVETOG.[16:1]#,%IF USING TSS FORTRAN.                        00169084 T  0001
SEGSW         = SAVETOG.[15:1]#,%TO MAINTAIN LINEDICT/LINESEG.                00169085 T  0001
TSSMESTOG     = SAVETOG.[14:1]#,%TSSMES(ERRMES).                              00169086 T  0001
WARNED        = SAVETOG.[13:1]#,%IF TSSEDIT WAS EVER EVOKED.                  00169087 T  0001
SEGSWFIXED    = SAVETOG.[12:1]#,%IF SEGSW IS NOT TO BE ALTERED.               00169088 T  0001
REMFIXED      = SAVETOG.[11:1]#,%IF REMOTETOG ISNT TO BE ALTERED.             00169089 T  0001
REMOTETOG     = SAVETOG.[10:1]#,%PRINT & READ STMTS = REMOTE.                 00169090 T  0001
RANDOMTOG     = SAVETOG.[ 9:1]#,%IF EXPR USED FOR RANDOM I/O.                 00169091 T  0001
VOIDTTOG      = SAVETOG.[ 8:1]#,%TO VOID TAPE RECORDS ONLY.                   00169092 T  0001
DOLIST        = SAVETOG.[ 7:1]#,%LIST DOLLAR CARDS.                           00169093 T  0001
HOLTOG        = SAVETOG.[ 6:1]#,%INPUT IN HOLLERITH.                          00169094 T  0001
LISTPTOG      = SAVETOG.[ 5:1]#,%LIST PATCHES ONLY.                           00169095 T  0001
PXREF         = SAVETOG.[ 4:1]#,%TO LIST XREF OF PROGRAM.                     00169096 T  0001
LASTLINE      = SAVETOG.[ 3:1]#,%TO DETR TO SKIP XREF.                        00169097 T  0001
XGLOBALS      = SAVETOG.[ 2:1]#,%TO LET XREF KNOW OF DOING GLOBAL             00169098 T  0001
NTAPTOG       = SAVETOG.[ 1:1]#,%TO CLOSE NEWTAPE IF EVER OPENED              00169099 T  0001
SEENADOUB     = SAVETOG.[40:1]#,%ARRAYDEC WONT BE USING THIS                  00169100 T  0001
                                %WHILE WE FIX DP COMMON ARRAY SZ              00169102 T  0001
                ENDSAVTOGDEF=#;                                               00169199 T  0001
                ARRAY SSNM[0:18] ; % THESE WORDS ARE USED FOR TEMP STORAGE AND STORING  00169200 T  0001
PRT(133) = SSNM
                                % PERMANENT FLAGGED DATA.                     00169201 T  0003
                DEFINE LASTSEQ=SSNM[0]#, LASTERR=SSNM[1]#, LINKLIST=SSNM[2]# ;  00169210 T  0003
                DEFINE VOIDSEQ=SSNM[3] # ;                                    00169220 T  0003
                ALPHA ARRAY ERRORBUFF,PRINTBUFF[0:14] ;                       00169300 T  0003
PRT(134) = ERRORBUFF
PRT(135) = PRINTBUFF
                ARRAY INLINEINT[0:35] ;                                       00169310 T  0005
PRT(136) = INLINEINT
                DEFINE XRBUFF = 150#, XRBUFFDIV3=50 #;                        00169320 T  0007
                ARRAY XRRY[0:XRBUFF-1] ;                                      00169330 T  0007
PRT(137) = XRRY
                INTEGER SEQBASE,SEQINCR;                                      00170000 T  0010
PRT(140) = SEQBASE
PRT(141) = SEQINCR
                INTEGER SCN,LABL,NEXTSCN ,NEXTCARD;                           00174000 T  0010
PRT(142) = SCN
PRT(143) = LABL
```

```
PRT(144) = NEXTSCN
PRT(145) = NEXTCARD
                INTEGER  SAVECARD;      % TO SAVE NEXTCART WHILE IN $ INCLUDE          00174100 T  0010
PRT(146) = SAVECARD
                INTEGER RESULT,INSERTDEPTH;                                           00174200 T  0010
PRT(147) = RESULT
PRT(150) = INSERTDEPTH
                REAL    INCLUDE;  % CONTAINS "INCLUDE"                                00174300 T  0010
PRT(151) = INCLUDE
                REAL DEBUGADR ;                                                       00174310 T  0010
PRT(152) = DEBUGADR
                ALPHA ACRO, CHRO, INITIALNCR;                                         00175000 T  0010
PRT(153) = ACRO
PRT(154) = CHRO
PRT(155) = INITIALNCR
                ALPHA ACR, NCR;                                                       00176000 T  0010
PRT(156) = ACR
PRT(157) = NCR
                SAVE ARRAY TYPE[0:63] ;                                               00176500 T  0010
PRT(160) = TYPE
                REAL LASTNEXT ;                                                       00176502 T  0012
PRT(161) = LASTNEXT
                ALPHA NEXTACC, NEXTACC2;                                              00177000 T  0012
PRT(162) = NEXTACC
PRT(163) = NEXTACC2
                ALPHA BLANKS;                                                         00178000 T  0012
PRT(164) = BLANKS
                ALPHA XTA;                                                            00179000 T  0012
PRT(165) = XTA
                ALPHA ARRAY EDOC[0:7, 0:127];   COMMENT HOLDS CODE EMITTED;           00180000 T  0012
PRT(166) = EDOC
                ALPHA ARRAY HOLDID[0:2];                                              00181000 T  0014
PRT(167) = HOLDID
                REAL NXAVIL,STRTSEG;                                                  00182000 T  0016
PRT(170) = NXAVIL
PRT(171) = STRTSEG
                ALPHA ARRAY WOP[0:139];   % HOLDS NAME AND CODE OF WORD MODE OPERATORS 00183000 T  0016
PRT(172) = WOP
                SAVE ALPHA ARRAY DB,TB,CB[0:9],                                       00184000 T  0017
PRT(173) = DB
PRT(174) = TB
PRT(175) = CB
                                CRD[0:14];                                            00185000 T  0020
PRT(176) = CRD
                ALPHA ARRAY XR[0:32];  INTEGER XRI;                                   00185100 T  0021
PRT(177) = XR
PRT(200) = XRI
                SAVE ARRAY ACCUM, EXACCUM[0:20] ;                                     00186000 T  0023
PRT(201) = ACCUM
PRT(202) = EXACCUM
                REAL ACCUMSTOP, EXACCUMSTOP ;                                         00186100 T  0025
PRT(203) = ACCUMSTOP
PRT(204) = EXACCUMSTOP
                REAL DBLOW;                                                           00187000 T  0025
PRT(205) = DBLOW
                REAL MAX;                                                             00188000 T  0025
PRT(206) = MAX
```

```
                ALPHA ACR1, CHR1;                                                   00189000 T  0025
PRT(207) = ACR1
PRT(210) = CHR1
                ALPHA ARRAY PERIODWORD[0:10];                                       00190000 T  0025
PRT(211) = PERIODWORD
                REAL ARRAY MAP[0:7];                                                00191000 T  0027
PRT(212) = MAP
                REAL F1, F2;                                                        00192000 T  0029
PRT(213) = F1
PRT(214) = F2
                INTEGER C1, C2;                                                     00193000 T  0029
PRT(215) = C1
PRT(216) = C2
                DEFINE                                                              00194000 T  0029
                        RSP=14 #, RSP1=15 #, RWP=29 #,                              00194020 T  0029
                        RSH=41 #, RSH1=42 #, RWS=83 #;                              00194040 T  0029
                ALPHA ARRAY RESERVEDWORDSLP[0:RWP], RESLENGTHLP,LPGLOBAL[0:RSP],    00195000 T  0029
PRT(217) = RESERVEDWORDSLP
PRT(220) = RESLENGTHLP
PRT(221) = LPGLOBAL
                        RESERVEDWORDS  [0:RWS], RESLENGTH          [0:RSH] ;        00195010 T  0033
PRT(222) = RESERVEDWORDS
PRT(223) = RESLENGTH
                SAVE REAL ARRAY PR, OPST [0:25];  % USED IN EXPR ONLY               00196000 T  0036
PRT(224) = PR
PRT(225) = OPST
                DEFINE PARMLINK = LSTT#;                                            00197000 T  0038
                ALPHA BUFF, BUFL, SYMBOL;                                           00198000 T  0038
PRT(226) = BUFF
PRT(227) = BUFL
PRT(230) = SYMBOL
                INTEGER  TV, % INDEX INTO INFO FOR PRT OF LIST NAMES                00199100 T  0038
PRT(231) = TV
                         SAVESUBS, %  MAX NUMBER OF SUBSCRIPTS FOR NAMELIST IDENTIFIERS  00199120 T  0038
PRT(232) = SAVESUBS
                         NAMEIND;  %  INDEX INTO NAMLIST IDENIFIERS ARRAY           00199130 T  0038
PRT(233) = NAMEIND
                ARRAY    NAMLIST[0:256]; % ARRAY TO STOKE NAMELIST IDENTIFIERS     00199200 T  0038
PRT(234) = NAMLIST
                ALPHA    LISTID;                                                    00199300 T  0040
PRT(235) = LISTID
                REAL ARRAY BDPRT[0:20]; REAL BDX;                                   00200000 T  0040
PRT(236) = BDPRT
PRT(237) = BDX
                DEFINE MAXSTRING = 49#;                                             00201000 T  0041
                ALPHA ARRAY STRINGARRAY[0:MAXSTRING];                               00202000 T  0041
PRT(240) = STRINGARRAY
                REAL STRINGSIZE;                  % NUMBER OF WORDS IN STRING        00203000 T  0043
PRT(241) = STRINGSIZE
                DEFINE LSTMAX = 256#;                                               00204000 T  0043
                REAL ARRAY LSTT, LSTP[0:LSTMAX];                                    00205000 T  0043
PRT(242) = LSTT
PRT(243) = LSTP
                INTEGER LSTI,LSTS,LSTA;                                             00206000 T  0045
PRT(244) = LSTI
PRT(245) = LSTS
PRT(246) = LSTA
```

```
                DEFINE MAXOPFILES = 63#, BIGGESTFILENB = 99#;                                   00207000 T  0045
                ARRAY FILEINFO[0:3,0:MAXOPFILES];                                               00208000 T  0045
PRT(247) = FILEINFO
                DEFINE    %     SOME FILE STUFF                                                 00208100 T  0047
                          SLOWV      = 2#,                                                      00208110 T  0047
                          FASTV      = 1#,                                                      00208120 T  0047
                          SENSPDEUNF= [1:8]#,                                                   00208130 T  0047
                               SENSF= [1:1]#,                                                   00208135 T  0047
                               SPDF = [2:2]#,                                                   00208140 T  0047
                               EUNF = [4:5]#,                                                   00208150 T  0047
                          FPBVERSION= 1#,                                                       00208160 T  0047
                          FPBVERSF   = [1:8]#,                                                  00208170 T  0047
                          DKAREASZ   = [9:39]#,                                                 00208180 T  0047
                ENDFILDEF=#;                                                                    00208490 T  0047
                INTEGER MAXFILES;                                                               00209000 T  0047
PRT(250) = MAXFILES
                ARRAY TEN[0:137];                                                               00210000 T  0047
PRT(251) = TEN
                DEFINE LBRANCH = 50#;                                                           00211000 T  0049
                REAL ARRAY BRANCHES[0:LBRANCH]; REAL LAX, BRANCHX;                              00212000 T  0049
PRT(252) = BRANCHES
PRT(253) = LAX
PRT(254) = BRANCHX
                INTEGER INXFIL,FILEARRAYPRT;                                                    00213000 T  0051
PRT(255) = INXFIL
PRT(256) = FILEARRAYPRT
                DEFINE MAXCOM=15 # ;                                                            00214000 T  0051
                INTEGER SUPERMAXCOM; %%% SUPERMAXCOM = 128*(MAXCOM+1),                          00214100 T  0051
PRT(257) = SUPERMAXCOM
                ALPHA ARRAY COM[0:MAXCOM,0:127] ;                                               00215000 T  0051
PRT(260) = COM
                REAL NEXTCOM;                                                                   00216000 T  0053
PRT(261) = NEXTCOM
                ALPHA ARRAY INFO[0:31, 0:127];                                                  00217000 T  0053
PRT(262) = INFO
                REAL ARYSZ;                                                                     00218000 T  0055
PRT(263) = ARYSZ
                ALPHA ARRAY EXTRAINFO[0:31, 0:127];                                             00219000 T  0055
PRT(264) = EXTRAINFO
                REAL EXPT1, EXPT2, EXPT3;                                                       00220000 T  0057
PRT(265) = EXPT1
PRT(266) = EXPT2
PRT(267) = EXPT3
                DEFINE IR = [36:5]#, IC = [41:7]#;                                              00221000 T  0057
                REAL INFA,INFB,INFC;                                                            00222000 T  0057
PRT(270) = INFA
PRT(271) = INFB
PRT(272) = INFC
                INTEGER NEXTINFO, GLOBALNEXTINFO, NEXTEXTRA;                                    00223000 T  0057
PRT(273) = NEXTINFO
PRT(274) = GLOBALNEXTINFO
PRT(275) = NEXTEXTRA
                INTEGER NEXTSS;                                                                 00224000 T  0057
PRT(276) = NEXTSS
                DEFINE SHX=189#, GHX=61#;                                                       00225000 T  0057
                REAL ARRAY STACKHEAD[0:SHX];                                                    00226000 T  0057
PRT(277) = STACKHEAD
```

```
                REAL ARRAY GLOBALSTACKHEAD[0:GHX];                                                  00227000 T  0058
PRT(300) = GLOBALSTACKHEAD
                REAL LA, DT, TEST;                                                                  00228000 T  0060
PRT(301) = LA
PRT(302) = DT
PRT(303) = TEST
                ALPHA LADR1,LADR2,LADR3,LADR4,LADR5,LINFA;   INTEGER LINDX,LADDR;                    00229000 T  0060
PRT(304) = LADR1
PRT(305) = LADR2
PRT(306) = LADR3
PRT(307) = LADR4
PRT(310) = LADR5
PRT(311) = LINFA
PRT(312) = LINDX
PRT(313) = LADDR
                DEFINE MAXDOS=20#;                                                                  00230000 T  0060
                ALPHA ARRAY DOTEST, DOLAB[0:MAXDOS];                                                00231000 T  0060
PRT(314) = DOTEST
PRT(315) = DOLAB
                ALPHA LISTART;                                                                      00232000 T  0062
PRT(316) = LISTART
                ALPHA ARRAY INT[0:NUMINTM1+NUMINTM1+2] ;                                            00233000 T  0062
PRT(317) = INT
                ALPHA ARRAY TYPES[0:6], KLASS[0:14];                                                00234000 T  0066
PRT(320) = TYPES
PRT(321) = KLASS
                INTEGER OP, PREC, IP, IT;                                                           00235000 T  0069
PRT(322) = OP
PRT(323) = PREC
PRT(324) = IP
PRT(325) = IT
                DEFINE DUMPSIZE =127 #;                                                             00237000 T  0069
                ARRAY FNNHOLD[0:DUMPSIZE];                                                          00238000 T  0069
PRT(326) = FNNHOLD
                INTEGER FNNINDEX,FNNPRT;                                                            00239000 T  0071
PRT(327) = FNNINDEX
PRT(330) = FNNPRT
                DEFINE MAXNBHANG = 30#;                                                             00240000 T  0071
                ARRAY FNNHANG[0:MAXNBHANG];                                                         00241000 T  0071
PRT(331) = FNNHANG
                DEFINE INTCLASS=[6:3]#, INTPARMCLASS=[9:3]#, INTINLINE=[12:6]#,                     00241010 T  0073
                       INTPRT =[24:6]#, INTPARMS   =[30:6]#, INTNUM  =[36:12]#,                     00241020 T  0073
                       INAM  =[12:36]#, INTX       =[2:10]#, INTSEEN   =[2:1]# ;                   00241030 T  0073
                DEFINE                                                                              00241200 T  0073
                       INSERTMAX = 20 #,  % MAX NO OF RECURSIVE CALL ALLOW ON LIB ROUT              00241220 T  0073
                       INSERTOP = INSERTINFO[INSERTDEPTH,4] #,                                      00241240 T  0073
                       INSERTMID = INSERTINFO[INSERTDEPTH,0] #,                                     00241260 T  0073
                       INSERTFID = INSERTINFO[INSERTDEPTH,1] #,                                     00241280 T  0073
                       INSERTINX = INSERTINFO[INSERTDEPTH,2] #,                                     00241300 T  0073
                       INSERTSEQ = INSERTINFO[INSERTDEPTH,3] #;                                     00241320 T  0073
                ARRAY INSERTINFO[0:INSERTMAX,0:4];                                                  00241360 T  0073
PRT(332) = INSERTINFO
                DEFINE MSRW=11 #;                                                                   00241900 T  0075
                ALPHA ARRAY MESSAGE[0:MSRW,0:96] ;                                                  00242000 T  0075
PRT(333) = MESSAGE
                BOOLEAN ARRAY MSFL[0:MSRW] ;                                                        00242050 T  0077
PRT(334) = MSFL
```

```
                ALPHA ARRAY LINEDICT[0:7,0:127];  % LINE DICTIONARY ARRAY                    00242100 T  0079
PRT(335) = LINEDICT
                ALPHA ARRAY LINESEG [0:11,0:127];  % LINE SEGMENT ARRAY                     00242200 T  0081
PRT(336) = LINESEG
                ARRAY TSSMESA[0:15] ; %%% BIT FLAGS PRNTD ERR MESSGS (≤748).              00242220 T  0083
PRT(337) = TSSMESA
                INTEGER NOLIN;  % NUMBER OF ENTRIES IN LINE SEGMENT                       00242300 T  0084
PRT(340) = NOLIN
                REAL LASTADDR ; %%% STORES LAST ADR.                                      00242700 T  0084
PRT(341) = LASTADDR
                INTEGER WARNCOUNT ; %%% COUNTS NUMBER OF TSS WARNINGS.                    00242750 T  0084
PRT(342) = WARNCOUNT
                DEFINE CE       = [2:1]#,        %INFA                    %996-          00243000 P  0084
                       LASTC    = [3:12]#,       %INFA                    %996-          00244000 P  0084
                       SEGNO    = [3:9]#,        %INFA                    %996-          00245000 P  0084
                       CLASS    = [15:5]#,       %INFA                    %996-          00246000 P  0084
                       EQ       = [20:1]#,       %INFA                    %996-          00246100 P  0084
                       SUBCLASS = [21:3]#,       %INFA                    %996-          00247000 P  0084
                       CLASNSUB = [14:10]#,      %INFA                    %996-          00248000 P  0084
                       ADJ      = [14:1]#,       %INFA                    %996-          00249000 P  0084
                       TWOD     = [14:1]#,       %INFA                    %996-          00250000 P  0084
                       FORMAL   = [13:1]#,       %INFA                    %996-          00251000 P  0084
                       TYPEFIXED= [12:1]#,       %INFA                    %996-          00252000 P  0084
                       ADDR     = [24:12]#,      %INFA                    %996-          00253000 P  0084
                       LINK     = [36:12]#,      %INFC                    %996-          00254000 P  0084
                       BASE     = [18:15]#,      %INFC                    %996-          00255000 P  0084
                       SIZE     = [33:15]#,      %INFC                    %996-          00256000 P  0084
                       BASENSIZE= [18:30]#,      %INFC                    %996-          00257000 P  0084
                       NEXTRA   = [2:6]#,        %INFC                    %996-          00258000 P  0084
                       ADINFO   = [8:10]#,       %INFC                    %996-          00259000 P  0084
                       RELADD   = [21:15]#,      %INFA                    %996-          00260000 P  0084
                       TOCF     = 2:47:1#,       %INFA                    %996-          00261000 P  0084
                       TOSEGNO  = 3:39:9#,       %INFA                    %996-          00262000 P  0084
                       TOLASTC  = 3:36:12#,      %INFA                    %996-          00262100 P  0084
                       TOCLASS  = 15:43:5#,      %INFA                    %996-          00263000 P  0084
                       TOEQ     = 20:47:1#,      %INFA                    %996-          00263100 P  0084
                       TOSUBCL  = 21:45:3#,      %INFA                    %996-          00264000 P  0084
                       TOADJ    = 14:47:1#,      %INFA                    %996-          00265000 P  0084
                       TOFORMAL = 13:47:1#,      %INFA                    %996-          00266000 P  0084
                       TOTYPE   = 12:47:1#,      %INFA                    %996-          00267000 P  0084
                       TOADDR   = 24:36:12#,     %INFA                    %996-          00268000 P  0084
                       TOLINK   = 36:36:12#,     %INFC                    %996-          00269000 P  0084
                       TOBASE   = 18:33:15#,     %INFC                    %996-          00270000 P  0084
                       TOSIZE   = 33:33:15#,     %INFC                    %996-          00271000 P  0084
                       TONEXTRA = 2:42:6#,       %INFC                    %996-          00272000 P  0084
                       TOADINFO = 8:38:10#,      %INFC                    %996-          00273000 P  0084
                       TORELADD = 21:33:15#,     %INFA                    %996-          00274000 P  0084
                %THE FOLLOWING CLASSES APPEAR IN THE 1ST WORD OF EACH 3-WORD ENTRY       00274998 C  0084
                %OF THE "INFO" ARRAY, AND MAY BE COPIED TO INFA.CLASS.          %996-    00274999 C  0084
                   UNKNOWN  = 0#,                                                        00275000 T  0084
                   ARRAYID  = 1#,                                                        00276000 T  0084
                   VARID    = 2#,                                                        00277000 T  0084
                   STMTFUNID= 3#,                                                        00278000 T  0084
                   NAMELIST = 4#,                                                        00279000 T  0084
                   FORMATID = 5#,                                                        00280000 T  0084
                   LABELID  = 6#,                                                        00281000 T  0084
                   FUNID    = 7#,                                                        00282000 T  0084
```

```
          ,NTRFUNID= 8#,                                                                    00283000 T  0084
          FXTID    = 9#,                                                                    00284000 T  0084
          SUBRID   = 10#,                                                                   00285000 T  0084
          BLOCKID  = 11#,                                                                   00286000 T  0084
          FILEID   = 12#,                                                                   00287000 T  0084
          SUBSVAR  = 13#,                                                                   00288000 T  0084
          EXPCLASS = 14#,                                                                   00289000 T  0084
          SBVEXP   = 15#,                                                                   00290000 T  0084
          LISTSID  = 16#,                                                                   00290050 T  0084
          JUNK     = 17#,          %NOT A GENUINE CLASS                    %996-            00290055 C  0084
          DUMMY    = 27#,                                                                   00290100 T  0084
          NUMCLASS = 28#,                                                                   00291000 T  0084
          RESERVED = 29#,                                                                   00292000 T  0084
          HEADER   = 30#,                                                                   00293000 T  0084
          ENDCOM   = 31#,                                                                   00294000 T  0084
      %THE FOLLOWING SUBCLASSES APPEAR IN THE 1ST WORD OF SOME 3-WORD%996-                   00294998 C  0084
      %ENTRIES OF THE "INFO" ARRAY, AND MAY BE COPIED TO INFA.SUBCLASS.                      00294999 C  0084
          ,NTYPE   = 1#,                                                                    00295000 T  0084
          STRINGTYPE=2#,                                                                    00296000 T  0084
          REALTYPE = 3#,                                                                    00297000 T  0084
          LOGTYPE  = 4#,                                                                    00298000 T  0084
          DOUBTYPE = 5#,                                                                    00299000 T  0084
          COMPTYPE = 6#;                                                                    00300000 T  0084
DEFINE    EOF= 500#, ID= 501#, NUM= 502#, PLUS= 503#, MINUS= 504#,                          00301000 T  0084
          STAR= 505#, SLASH= 506#, LPAREN= 507#, RPAREN= 508#,                              00302000 T  0084
          EQUAL= 509#, COMMA= 510#, SEMI= 511#, DOLLAR=0#, UPARROW=0# ;                     00303000 T  0084
DEFINE THRU = STEP 1 UNTIL #;                                                   %996-       00316000 P  0084
DEFINE                                                                                      00317000 T  0084
          ADD = 16#, BBC = 22#, BBW = 534#, BFC = 38#, BFW = 550#                           00318000 T  0084
          ,CDC =168#, CHS =134#, COC =  40#, KOM =130#, DEL =  10#                          00319000 T  0084
          ,DUP =261#, EQUL=581#, GBC = 278#, GBW =790#, GEQL=  21#                          00320000 T  0084
          ,GFC =294#, GFW =806#, GRTR=  37#, IDV =384#, INX =  24#                          00321000 T  0084
          ,ISD =532#, ISN =548#, LEQL= 533#, LND = 67#, LNG =  19#                          00322000 T  0084
          ,LOD =260#, LOR = 35#, LQV = 131#, LESS=549#, MDS = 515#                          00323000 T  0084
          ,MKS = 72#, MUL = 64#, NEQL=  69#, NOP = 11#, PRL =  18#                          00324000 T  0084
          ,XRT = 12#, RDV =896#, RTN = 39#, RTS =167#, SND = 132#                           00325000 T  0084
          , SSN = 70#, SSP = 582#, STD = 68#, SUB = 48#, XCH = 133#                         00326000 T  0084
          ,XIT = 71#, ZP1 =322#, DIU =128#, STN =132#                                       00327000 T  0084
          ,DIA = 45#, DIB = 49#, TRB = 53#, ISO = 37#                                       00328000 T  0084
          ,AD2 =17#, SB2 = 49#, ML2 = 65#, DV2 =129#, FTC = 197#                            00329000 T  0084
          ,SSF =280#, FTF =453#, CTC =709#, CTF = 965#                                      00330000 T  0084
          ,TOP=262#                                                                         00330010 T  0084
          ;                                                                                 00331000 T  0084
     FORMAT FD(X100,"NEXT = ",I3,X2,2A6) ;                                                  00332000 T  0084
                                                                    START OF SEGMENT **********     3
PRT(343) = FD

                                                                        3 IS    9 LONG, NEXT SEG    2
     FORMAT SEGSTRT(X63, " START OF SEGMENT **********", I4),                           00333000 T  0084
                                                                    START OF SEGMENT **********     4
PRT(344) = SEGSTRT
          FLAGROUTINEFORMAT(X41,A6,A2,X1,2A6," (",A1,I*,")"),                               00333010 T  0084
          XHEDM(X40,"CROSS REFERENCE LISTING OF MAIN PROGRAM",X41,/,                         00333050 T  0084
               X40,"----- -------- ------- -- ---- -------",X41),                          00333055 T  0084
          XHEDS(X38, "CROSS REFERENCE LISTING OF SUBROUTINE ",A6,X38,/,                     00333060 T  0084
               X38, "----- -------- ------- -- ---------- ",A6,X38),                       00333065 T  0084
          XHEDF(X35,"CROSS REFERENCE LISTING OF ",A6,A1," FUNCTION ",A6,                    00333070 T  0084
          X35,/,X35,"----- -------- ------- -- ",A6,A1," -------- ",A6,                    00333075 T  0084
```

```
                    X35),
                    XHEDG(X43,"CROSS REFERENCE LISTING OF GLOBALS",X43,/,                                    00333076 T  0084
                          X43,"----- --------- ------- -- --------",X43),                                   00333080 T  0084
                    XHEDB(X34,"CROSS REFERENCE LISTING OF BLOCK DATA SUBPROGRAM",X35,                        00333085 T  0084
                      /,X34,"----- --------- ------- -- ----- ---- ----------",X35),                        00333090 T  0084
                    SEGEND (X70," SEGMENT",I5," IS",I5," LONG");                                            00333095 T  0084
                                                                                                            00334000 T  0084
                                                                                          4 IS  150 LONG, NEXT SEG    2
               ARRAY PDPRT [0:31,0:63];                                                                     00335000 T  0084
PRT(345) = PDPRT
               REAL PDINX;        % INDEX OF LAST ENTRY IN PDPRT                                         00336000 T  0086
PRT(346) = PDINX
               REAL TSEGSZ;       % RUNNING COUNT OF TOTAL SEGMENT SIZE;                                 00337000 T  0086
PRT(347) = TSEGSZ
                  COMMENT FORMAT OF PRT&SEGMENT ENTRIES, IN PDPRT;                                          00338000 T  0086
               DEFINE       STYPF= [1:2]#,  %  0 = PROGRAM SEGMENT-SEGMENT ENTRY ONLY                    00339000 T  0086
                            STYPC= 1:46:2#, %  1 = MCP INTRINSIC                                         00340000 T  0086
                                            %  2 = DATA SEGMENT                                          00341000 T  0086
                            DTYPF= [4:2]#,  %  0 = THUNK - PRT ENTRY ONLY                                00342000 T  0086
                            DTYPC= 4:46:2 #,%  1 = WORD MODE PROGRAM DESCRIPTOR                          00343000 T  0086
                                            %  2 = LABEL DESCRIPTOR                                      00344000 T  0086
                                            %  3 = CHARACTER MODE PROGRAM DESCRIPTOR                     00345000 T  0086
                            PRTAF= [8:10]#, %  ACTUAL PRT ADDRESS - PRT ENTRY                            00346000 T  0086
                            PRTAC= 8:38:10#,                                                             00347000 T  0086
                            RELADF = [18:10]#, % ADDRESS WITHIN SEGMENT -PRT ONLY                        00348000 T  0086
                            RELADC= 18:38:10#,                                                           00349000 T  0086
                            SGNOF=  [28:10]#, % SEGMENT NUMBER - BOTH                                    00350000 T  0086
                            SGNOC= 28:38:10#,                                                            00351000 T  0086
                            DKAI = [13:15]#,  % RELATIVE DISK ADDRESS - SEGMENT ENTRY                    00352000 T  0086
                            DKAC = 13:33:15#,                                                            00353000 T  0086
                            SEGSZF =[38:10]#, % SEGMENT SIZE - SEGMENT ENTRY                             00354000 T  0086
                            SEGSZC =38:38:10#;% MUST BE 0 IF PRT ENTRY                                   00355000 T  0086
                     DEFINE %       SOME EXTERNAL CONSTANTS                                              00355100 T  0086
                               BLKCNTRLINT     = 5#,                                                     00355110 T  0086
                               FPLUS2          = 1538#,                                                  00355120 T  0086
                     ENDEXTCONDEF=#;                                                                     00355990 T  0086
                     DEFINE    PDIR = PDINX,[37:5]#,                                                     00356000 T  0086
                               PDIC = PDINX,[42:6]#;                                                     00357000 T  0086
                     DEFINE CIS = CRD[0]#,    % CARD                                                     00358000 T  0086
                            TIS = CRD[1]#,    % TAPE                                                     00359000 T  0086
                            TOS = CRD[2]#,    % NEW TAPE                                                 00360000 T  0086
                            LOS = CRD[3]#;    % PRINTER                                                  00361000 T  0086
                     DEFINE PWROOT=ROOT,IR,ROOT,IC #,PWI=I,IR,I,IC # ;                                   00362000 T  0086
                     DEFINE GLOBALNEXT=NEXT#;                                                            00363000 T  0086
                     BEGIN    % SCAN FPB TO SEE IF VARIOUS FILES ARE DISK OR TAPE                        00364000 T  0086
                     INTEGER STREAM PROCEDURE GETFPB(Q); VALUE Q;                                        00365000 T  0086
PRT(350) = *SEGMENT DESCRIPTOR*
                                                                                  START OF SEGMENT **********    5
PRT(351) = GETFPB
                         BEGIN                                                                           00366000 T  0000
                              SI ← LOC GETFPB; SI ← SI + 7; DI ← LOC Q; DI ← DI + 5;                     00367000 T  0000
                              SKIP 3 DB; 9(IF SB THEN DS ← SET ELSE DS ← RESET;                          00368000 T  0001
                                           SKIP SB);                                                     00369000 T  0002
                              DI ← LOC Q; SI ← Q; DS ← WDS; SI ← Q; GETFPB ← SI;                          00370000 T  0003
                         END;                                                                            00371000 T  0004
                     INTEGER STREAM PROCEDURE GNC(Q); VALUE Q;                                           00372000 T  0005
PRT(352) = GNC
                         BEGIN SI ← Q; SI ← SI + 7; DI ← LOC GNC; DI ← DI +7; DS ←CHR END;               00373000 T  0005
```

```
                    INTEGER  FPBBASE;                                                                    00374000 T  0008
PRT(353) = FPBBASE
                    FPBBASE ← GETFPB(3);                                                                 00375000 T  0008
                    TIS ← TOS ← 50; CIS ← LOS ← 0;                                                       00376000 T  0010
                    IF GNC(FPBBASE+3) =12 THEN CIS ← 150;    % CARD                                       00377000 T  0015
                    IF GNC(FPBBASE+8) =12 THEN LOS ← 150;    % PRINTER;                                  00378000 T  0019
                    IF GNC(FPBBASE+13) = 12 THEN TOS ← 150;   % NEW TAPE;                               00379000 T  0023
                    IF GNC(FPBBASE+18) =12 THEN TIS ← 150;  % TAPE                                      00380000 T  0027
                    END;                                                                                 00381000 T  0031
PRT(354) = *SEGMENT DESCRIPTOR*
                                                                                    5 IS   32 LONG, NEXT SEG    2
                    BEGIN  COMMENT  INNER BLOCK;                                                         00382000 T  0088
                    % *** DO NOT DECLARE ANY FILES PRIOR TO THIS POINT, DO NOT ALTER                      00383000 T  0088
                    %      SEQUENCE OF FOLLOWING FILE DECLARATIONS                                       00384000 T  0088
                    FILE CARD (5,10,CIS);                                                                00385000 T  0088
PRT(355) = *SEGMENT DESCRIPTOR*
                                                                                 START OF SEGMENT **********    6
PRT(353) = CARD
                    DEFINE LINESIZE = 900 # ;                                                            00386000 T  0004
                    SAVE FILE LINE DISK SERIAL [20:LINESIZE] (2,15,LOS,SAVE 10);                         00387000 T  0004
PRT(356) = LINE
PRT(357) =
PRT(360) = FILE ATTRBUTS
                    SAVE FILE NEWTAPE DISK SERIAL [20:LINESIZE] "FORSYM" (2,10,TOS,SAVE 10);             00388000 T  0011
PRT(361) = NEWTAPE
                    FILE TAPE "FORSYM" (2,10,TIS);                                                       00389000 T  0019
PRT(362) = TAPE
                    FILE REMOTE 19(2,10) ;                                                               00389500 T  0023
PRT(363) = REMOTE
                    DEFINE CHUNK =180#;                                                                  00390000 T  0027
                    DEFINE PTR=LINE#,RITE=LINE#,CR=CARD#,TP=TAPE#;                          %515-        00391000 P  0027
                    FILE CODE DISK RANDOM [20:CHUNK]  (4,30,SAVE ABS(SAVETIME));                         00392000 T  0027
PRT(364) = CODE
                    FILE LIBRARYFIL DISK RANDOM (2,10,150);                                              00392400 T  0035
PRT(365) = LIBRARYFIL
                    DEFINE LF = LIBRARYFIL#;                                                             00392600 T  0039
                    FILE XREFF DISK SERIAL[20:1500](2,XRBUFF) ;                                          00392700 T  0039
PRT(366) = XREFF
                    FILE XREFG DISK SERIAL[20:1500](2,3,XRBUFF);                                         00392800 T  0043
PRT(367) = XREFG
                    REAL DALOC;         % DISK ADDRESS                                                  00393000 T  0047
PRT(370) = DALOC
                    LABEL POSTWRAPUP;                                                                    00393100 T  0047
                    PROCEDURE EMITNUM(N); VALUE N; REAL N; FORWARD;                                      00394000 T  0047
PRT(371) = EMITNUM
                    REAL PROCEDURE LOOKFORINTRINSIC(L);  VALUE L;  REAL L;  FORWARD ;                    00394500 T  0051
PRT(372) = LOOKFORINTRINSIC
                    PROCEDURE SEGOVF; FORWARD;                                                           00395000 T  0051
PRT(373) = SEGOVF
                    DEFINE BUMPADR= IF (ADR←ADR+1)=4089 THEN SEGOVF#;                                    00396000 T  0051
                    DEFINE BUMPLOCALS = BEGIN IF LOCALS←LOCALS+1>255 THEN BEGIN FLAG(148);               00396500 T  0051
                                        LOCALS←2 END END#;                                               00396510 T  0051
                    DEFINE BUMPPRT=BEGIN IF PRTS←PRTS+1>1023 THEN BEGIN FLAG(46);                        00396600 T  0051
                                        PRTS←40 END END#;                                                00396610 T  0051
                    DEFINE FDOCI = ADR.[36:3], ADR.[39:7]#;                                              00397000 T  0051
                    DEFINE TWODPRT=IF TWODPRTX=0 THEN(TWODPRTX←PRTS←PRTS+1)ELSE TWODPRTX#;               00398000 T  0051
                    REAL PROCEDURE SEARCH(E); VALUE E; REAL E; FORWARD;                                  00399000 T  0051
```

```
PRT(374) = SEARCH
                PROCEDURE PRINTCARD; FORWARD;                                   00400000 T  0051
PRT(375) = PRINTCARD
                INTEGER PROCEDURE FIELD(X); VALUE X; INTEGER X; FORWARD ;       00400010 T  0051
PRT(376) = FIELD
                ALPHA PROCEDURE NEED(T, C); VALUE T, C; ALPHA T, C; FORWARD;    00401000 T  0051
PRT(377) = NEED
                ALPHA PROCEDURE GETSPACE(S); VALUE S; ALPHA S; FORWARD;         00402000 T  0051
PRT(400) = GETSPACE
                PROCEDURE EQUIV(R); VALUE R; REAL R; FORWARD;                   00403000 T  0051
PRT(401) = EQUIV
                PROCEDURE EXECUTABLE; FORWARD;                                  00403100 T  0051
PRT(402) = EXECUTABLE
                ALPHA PROCEDURE B2D(B); VALUE B; REAL B; FORWARD;               00404000 T  0051
PRT(403) = B2D
                PROCEDURE DEBUGWORD (N); VALUE N; REAL N;       FORWARD;        00405000 T  0051
PRT(404) = DEBUGWORD
                PROCEDURE EMITL(N); VALUE N; REAL N; FORWARD;                   00406000 T  0051
PRT(405) = EMITL
                PROCEDURE ADJUST;FORWARD;                                       00407000 T  0051
PRT(406) = ADJUST
                PROCEDURE DATIME; FORWARD;                                      00407500 T  0051
PRT(407) = DATIME
                PROCEDURE EMITB(A,C); VALUE A,C; REAL A; BOOLEAN C; FORWARD;    00408000 T  0051
PRT(410) = EMITB
                PROCEDURE FIXB(N); VALUE N; REAL N; FORWARD;                    00408100 T  0051
PRT(411) = FIXB
                PROCEDURE EMITO(N); VALUE N; REAL N; FORWARD;                   00409000 T  0051
PRT(412) = EMITO
                PROCEDURE EMITOPDCLIT(N); VALUE N; REAL N; FORWARD;             00410000 T  0051
PRT(413) = EMITOPDCLIT
                PROCEDURE EMITDESCLIT(N); VALUE N; REAL N; FORWARD;             00411000 T  0051
PRT(414) = EMITDESCLIT
                PROCEDURE EMITPAIR(L,OP); VALUE L,OP; INTEGER L,OP; FORWARD;    00412000 T  0051
PRT(415) = EMITPAIR
                PROCEDURE EMITN(N); VALUE N; REAL N; FORWARD;                   00413000 T  0051
PRT(416) = EMITN
                PROCEDURE ARRAYDEC(I); VALUE I; REAL I; FORWARD;                00414000 T  0051
PRT(417) = ARRAYDEC
                INTEGER PROCEDURE ENTER(W, E); VALUE W, E; ALPHA W, E; FORWARD; 00415000 T  0051
PRT(420) = ENTER
                REAL PROCEDURE PRGDESCBLDR(A,B,C,D); VALUE A,B,C,D; REAL A,B,C,D;
                                                                                00416000 T  0051
PRT(421) = PRGDESCBLDR
                            FORWARD;                                            00417000 T  0051
                PROCEDURE WRITEDATA(A,B,C); VALUE A,B; REAL A,B;                00418000 T  0051
PRT(422) = WRITEDATA
                              ARRAY C[0];       FORWARD;                        00419000 T  0051
                PROCEDURE SCAN;                 FORWARD;                        00420000 T  0051
PRT(423) = SCAN
                PROCEDURE SEGMENT(A,B,C,D); VALUE A,B,C;                        00421000 T  0051
PRT(424) = SEGMENT
                        REAL A,B; BOOLEAN C; ARRAY D[0,0];    FORWARD;          00422000 T  0051
                STREAM PROCEDURE MOVESEQ(OLD,NEW);  BEGIN DI←OLD;  SI←NEW;  DS←WDS END ;
                                                                                00422010 T  0051
PRT(425) = MOVESEQ
                PROCEDURE WRITAROW(N,ROW); VALUE N; INTEGER N; REAL ARRAY ROW[0] ;
                                                                                00422100 T  0051
PRT(426) = WRITAROW
                IF SINGLETOG THEN WRITE(LINE,N,ROW[*]) ELSE WRITE(RITE,N,ROW[*]) ;
                                                                                00422110 T  0052
```

```
                PROCEDURE WRITALIST(FMT,N,L1,L2,L3,L4,L5,L6,L7,L8) ;                                    00422120 T  0062
PRT(427) = WRITALIST
                VALUE N,L1,L2,L3,L4,L5,L6,L7,L8; INTEGER N;                                             00422130 T  0062
                REAL L1,L2,L3,L4,L5,L6,L7,L8; FORMAT FMT ;                                              00422140 T  0062
                   BEGIN                                                                                00422150 T  0062
                   PRINTBUFF[1]←L1 ;                                                                    00422160 T  0062
                   L1←1 ;                                                                               00422170 T  0064
                   FOR L2←L2,L3,L4,L5,L6,L7,L8 DO PRINTBUFF[L1←L1+1]←L2 ;                               00422180 T  0065
STACK(F+2) = *TEMPORARY STORAGE*
                   IF SINGLETOG THEN WRITE(LINE,FMT,FOR L1←1 THRU N DO PRINTBUFF[L1])                   00422190 T  0082
PRT(430) = *LIST, LABEL, OR SEGMENT DESCRIPTOR*
                   ELSE WRITE(RITE,FMT,FOR L1←1 THRU N DO PRINTBUFF[L1]) ;                              00422200 T  0090
PRT(431) = OUTPUT(W)
PRT(432) = *LIST, LABEL, OR SEGMENT DESCRIPTOR*
                   END WRITALIST ;                                                                      00422210 T  0107
                STREAM PROCEDURE BLANKIT(A,N,P);  VALUE N,P;                                            00422220 T  0109
PRT(433) = BLANKIT
                BEGIN DI←A;  N(DS←8 LIT"        ");  P(DS←8 LIT"99999999") END ;                        00422230 T  0109
                BOOLEAN STREAM PROCEDURE TSSMES(A,B);  VALUE B;                                         00422235 T  0115
PRT(434) = TSSMES
                BEGIN SI←A;  SKIP B SB;  IF SB THEN ELSE BEGIN TALLY←1;  TSSMES←TALLY ;                 00422240 T  0115
                DI←A;  SKIP B DB;  DS←SET END END OF TSSMES ;                                           00422250 T  0118
                STREAM PROCEDURE TSSEDIT(X,P1,P2,P3,P,N);  VALUE P1,P2,P3,N ;                           00422255 T  0120
PRT(435) = TSSEDIT
                BEGIN DI←P;DS←16LIT"TSS WARNING:    ";SI←X;SI←SI+2; DS←6CHR; DS←4LIT" ";                00422260 T  0120
                P1(DS←48LIT"A TSS HOL OR QUOTED STRING MUST BE ON 1 LINE    ") ;                        00422262 T  0124
                P2(DS←48LIT"THIS CONSTRUCT IS ILLEGAL IN TSS FORTRAN        ") ;                        00422264 T  0131
                P3(DS←48LIT"THIS CONSTRUCT IS UNRESERVED IN TSS FORTRAN     ") ;                        00422266 T  0139
                DS←26LIT" ";  DS←8LIT"*";  DS←LIT" ";  SI←LOC N;  DS←3DEC END TSSEDIT ;                  00422268 T  0146
                PROCEDURE TSSED(X,N);  VALUE X,N;  ALPHA X;  INTEGER N ;                                00422270 T  0152
PRT(436) = TSSED
                BEGIN IF NOT (LISTOG OR SEQERRORS) THEN PRINTCARD ;  UNPRINTED←FALSE ;                  00422275 T  0152
                TSSEDIT(X,N=1,N=2,N=3,PRINTBUFF,WARNCOUNT←WARNCOUNT+1) ;                                00422280 T  0157
                WRITAROW(14,PRINTBUFF) END OF TSSED ;                                                   00422285 T  0162
                PROCEDURE ERRMESS(N);  VALUE N;  INTEGER N;                                             00423000 T  0164
PRT(437) = ERRMESS
                BEGIN                                                                                   00424000 T  0164
                                                                                                        00425000 T  0164
                STREAM PROCEDURE PLACE(D,N,X,S,E,L);  VALUE N,E ;                                       00426000 T  0164
                                                                                  START OF SEGMENT **********    7
PRT(440) = PLACE
                BEGIN DI ← D;  DS ← 6 LIT "ERROR ";                                                     00427000 T  0000
                      SI ← LOC N;  DS ← 3 DEC;  DS ← 5 LIT ":    ";                                     00428000 T  0001
                      SI ← X;  SI ← SI + 2;  DS ← 6 CHR;  DS ← 4 LIT " ";                               00429000 T  0002
                      SI←S;  DS←6 WDS;  DS←6 LIT" ";  SI←L;  DS←8 CHR;  DS←14 LIT" " ;                  00430000 T  0004
                      DS ← 8 LIT "X";  DS ← LIT " ";                                                    00431000 T  0008
                      SI ← LOC E;  DS ← 3 DEC;                                                          00432000 T  0010
                END PLACE;                                                                              00433000 T  0010
                STREAM PROCEDURE DCPLACE(D,N,X,S,M,P);  VALUE N,P ;                                     00433100 T  0010
PRT(441) = DCPLACE
                    BEGIN DI←D;  DS←4LIT"ERR#";  SI←LOC N;  DS←3 DEC;  DS←3LIT" @ " ;                   00433200 T  0010
                    SI←S; DS←8CHR; DS←2LIT":"; SI←X;  SI←SI+2;  DS←6CHR; DS←54LIT" " ;                  00433300 T  0013
                    P(DI←DI-54;  DS←2LIT"  ";  SI←M;  DS←44CHR;  DS←8LIT" ") ;                          00433400 T  0022
                    END OF DCPLACE ;                                                                    00433500 T  0025
                ERRORCT←ERRORCT+1 ;                                                                     00434000 T  0026
                IF NOT MSFLIN DIV 161 THEN                                                              00434010 T  0027
                    BEGIN                                                                               00434020 T  0028
```

```
                    MSFLIN DIV 16]←TRUE ;                                                                  00434030 T  0029
                    CASE(N DIV 16)OF                                                                       00434040 T  0030
                BEGIN                                                                                      00435000 T  0031
PRT(442) = *CASE STATEMENT DESCRIPTOR*
                    FILL MESSAGE[0,*] WITH                                                                 00436000 T  0032
                "SYNTAX E","RROR    ","         ","         ","         ", %000                           00437000 T  0032
                                                                                                START OF SEGMENT **********    8
                "MISSING ","OPERATOR","  OR PUNC","TUATION ","         ","         ", %001                  00438000 T  0034
                "CONFLICT","ING COMM","ON AND/O","R EQUIVA","LENCE AL","LOCATION", %002                    00439000 T  0034
                "MISSING ","RIGHT PA","RENTHESI","S        ","         ","         ", %003                  00440000 T  0034
                "ENTRY ST","MT ILLEG","AL IN MA","IN PGM O","R BLOCK ","DATA     ", %004                   00441000 T  0034
                "MISSING ","END STAT","EMENT    ","         ","         ","         ", %005                  00442000 T  0034
                "ARITHMET","IC EXPRE","SSION RE","QUIRED   ","         ","         ", %006                  00443000 T  0034
                "LOGICAL ","EXPRESSI","ON REQUI","RED      ","         ","         ", %007                  00444000 T  0034
                "TOO MANY","  LEFT PA","RENTHESE","S        ","         ","         ", %008                  00445000 T  0034
                "TOO MANY","  RIGHT P","ARENTHES","ES       ","         ","         ", %009                  00446000 T  0034
                "FORMAL P","ARAMETER","  ILLEGAL","  IN COMM","ON       ","         ", %010                  00447000 T  0034
                "FORMAL P","ARAMETER","  ILLEGAL","  IN EQUI","VALENCE ","         ", %011                  00448000 T  0034
                "THIS STA","TEMENT I","LLEGAL I","N BLOCK ","DATA SUB","PROGRAM ", %012                    00449000 T  0034
                "INFO ARR","AY OVERF","LOW      ","         ","         ","         ", %013                  00450000 T  0034
                "IMPROPER","  DO NEST","         ","         ","         ","         ", %014                 00451000 T  0034
                "DO LABEL","  PREVIOU","SLY DEFI","NED      ","         ","         ", %015                  00452000 T  0034
                0;                                                                                         00453000 T  0034
                                                                                                      8 IS   97 LONG, NEXT SEG    7
                    FILL MESSAGE[1,*] WITH                                                                 00454000 T  0034
                "UNRECOGN","IZED STA","TEMENT T","YPE      ","         ","         ", %016                  00455000 T  0035
                                                                                                START OF SEGMENT **********    9
                "ILLEGAL ","DO STATE","MENT     ","         ","         ","         ", %017                  00456000 T  0036
                "FORMAT S","TATEMENT","  MUST HA","VE LABEL","         ","         ", %018                  00457000 T  0036
                "UNDEFINE","D LABEL  ","         ","         ","         ","         ", %019                 00458000 T  0036
                "MULTIPLE","  DEFINIT","ION      ","         ","         ","         ", %020                  00459000 T  0036
                "ILLEGAL ","IDENTIFI","ER CLASS","  IN THIS","  CONTEXT","         ", %021                   00460000 T  0036
                "UNPAIRED","  QUOTES ","IN FORMA","T        ","         ","         ", %022                   00461000 T  0036
                "NOT ENOU","GH SUBSC","RIPTS    ","         ","         ","         ", %023                  00462000 T  0036
                "TOO MANY","  SUBSCRI","PTS      ","         ","         ","         ", %024                  00463000 T  0036
                "FUNCTION","  OR SUBR","OUTINE P","REVIOUSL","Y DEFINE","D        ", %025                   00464000 T  0036
                "FORMAL P","ARAMETER","  MULTIPL","Y DEFINE","D IN HEA","DING     ", %026                   00465000 T  0036
                "ILLEGAL ","USE OF N","AMELIST  ","         ","         ","         ", %027                  00466000 T  0036
                "NUMBER O","F PARAME","TERS INC","ONSISTEN","T        ","         ", %028                   00467000 T  0036
                "CANNOT B","RANCH TO","  FORMAT ","STATEMEN","T        ","         ", %029                   00468000 T  0036
                "SUBROUTI","NE OR FU","NCTION N","OT DEFIN","ED IN PR","OGRAM    ", %030                   00469000 T  0036
                "IDENTIFI","ER ALREA","DY GIVEN","  TYPE   ","         ","         ", %031                  00470000 T  0036
                0;                                                                                         00471000 T  0036
                                                                                                      9 IS   97 LONG, NEXT SEG    7
                    FILL MESSAGE[2,*] WITH                                                                 00472000 T  0037
                "ILLEGAL ","FORMAT S","YNTAX    ","         ","         ","         ", %032                  00473000 T  0037
                                                                                                START OF SEGMENT **********   10
                "INCORREC","T USE OF","  FILE   ","         ","         ","         ", %033                  00474000 T  0039
                "INCONSIS","TENT USE","  OF IDEN","TIFIER   ","         ","         ", %034                  00475000 T  0039
                "ARRAY ID","ENTIFIER","  EXPECTE","D        ","         ","         ", %035                  00476000 T  0039
                "EXPRESSI","ON VALUE","  REQUIRE","D        ","         ","         ", %036                  00477000 T  0039
                "ILLEGAL ","FILE CAR","D SYNTAX","         ","         ","         ", %037                  00478000 T  0039
                "ILLEGAL ","CONTROL ","ELEMENT ","         ","         ","         ", %038                   00479000 T  0039
                "DECLARAT","ION MUST","  PRECEDE","  FIRST R","EFERENCE","         ", %039                   00480000 T  0039
                "INCONSIS","TENT USE","  OF LABE","L AS PAR","AMETER   ","         ", %040                   00481000 T  0039
                "NO. OF P","ARAMS. D","ISAGREES","  WITH PR","EV. REFE","RENCE    ", %041                   00482000 T  0039
                "ILLEGAL ","USE OF F","ORMAL PA","RAMETER ","         ","         ", %042                   00483000 T  0039
```

```
"ERROR IN","  HOLLERI","TH LITER","AL CHARA","CTER COU","NT      ", %043          00484000 T  0039
"ILLEGAL ","ACTUAL P","ARAMETER","       ","       ","        ", %044          00485000 T  0039
"TOO MANY","  SEGMENT","S IN SOU","RCE PROG","RAM     ","      ", %045          00486000 T  0039
"TOO MANY","  PRT ASS","IGNMENTS","  IN SOUR","CE PROGR","AM    ", %046          00487000 T  0039
"LAST BLO","CK DECLA","RATION H","AD LESS ","THAN 102","4 WORDS ", %047          00488000 T  0039
0;                                                                               00489000 T  0039
                                                                          10 IS   97 LONG, NEXT SEG     7
        FILL MESSAGE[3,*] WITH                                                   00490000 T  0039
"ILLEGAL ","I/O LIST","  ELEMENT","       ","       ","        ", %048          00491000 T  0040
                                                                        START OF SEGMENT **********   11
"LEFT SID","E MUST B","E SIMPLE","  OR SUBS","CRIPTED ","VARIABLE", %049          00492000 T  0041
"VARIABLE","  EXPECTE","D       ","       ","       ","        ", %050          00493000 T  0041
"ILLEGAL ","USE OF .","OR.     ","       ","       ","        ", %051          00494000 T  0041
"ILLEGAL ","USE OF .","AND.    ","       ","       ","        ", %052          00495000 T  0041
"ILLEGAL ","USE OF .","NOT.    ","       ","       ","        ", %053          00496000 T  0041
"ILLEGAL ","USE OF R","ELATIONA","L OPERAT","OR     ","      ", %054          00497000 T  0041
"ILLEGAL ","MIXED TY","PES     ","       ","       ","        ", %055          00498000 T  0041
"ILLEGAL ","EXPRESSI","ON STRUC","TURE    ","       ","      ", %056          00499000 T  0041
"ILLEGAL ","PARAMETE","R       ","       ","       ","        ", %057          00500000 T  0041
"RECORD B","LOCK GRE","ATER THA","N 1023  ","       ","      ", %058          00501000 T  0041
"TOO MANY","  OPTIONA","L FILES ","       ","       ","        ", %059          00502000 T  0041
"FILE CAR","DS MUST ","PRECEDE ","SOURCE D","ECK     ","      ", %060          00503000 T  0041
"BINARY W","RITE STA","TEMENT H","AS NO LI","ST      ","      ", %061          00504000 T  0041
"UNDEFINE","D FORMAT","  NUMBER ","       ","       ","        ", %062          00505000 T  0041
"ILLEGAL ","EXPONENT","  IN CONS","TANT    ","       ","      ", %063          00506000 T  0041
0;                                                                               00507000 T  0041
                                                                          11 IS   97 LONG, NEXT SEG     7
        FILL MESSAGE[4,*] WITH                                                   00508000 T  0042
"ILLEGAL ","CONSTANT","  IN DATA","  STATEME","NT      ","      ", %064          00509000 T  0042
                                                                        START OF SEGMENT **********   12
"MAIN PRO","GRAM MIS","SING    ","       ","       ","        ", %065          00510000 T  0044
"PARAMETE","R MUST B","E ARRAY ","IDENTIFI","ER      ","      ", %066          00511000 T  0044
"PARAMETE","R MUST B","E EXPRES","SION    ","       ","      ", %067          00512000 T  0044
"PARAMETE","R MUST B","E LABEL ","       ","       ","        ", %068          00513000 T  0044
"PARAMETE","R MUST B","E FUNCTI","ON IDENT","IFIER   ","      ", %069          00514000 T  0044
"PARAMETE","R MUST B","E FUNCTI","ON OR SU","BROUTINE","  ID   ", %070          00515000 T  0044
"PARAMETE","R MUST B","E SUBROU","TINE IDE","NTIFIER ","      ", %071          00516000 T  0044
"PARAMETE","R MUST B","E ARRAY ","IDENTIFI","ER OR EX","PRESSION", %072          00517000 T  0044
"ARITHMET","IC - LOG","ICAL CON","FLICT ON","  STORE  ","      ", %073          00518000 T  0044
"ARRAYID ","MUST BE ","SUBSCRIP","TED IN T","HIS CONT","EXT     ", %074          00519000 T  0044
"MORE THA","N ONE MA","IN PROGR","AM      ","       ","      ", %075          00520000 T  0044
"ONLY COM","MON ELEM","ENTS PER","MITTED  ","       ","      ", %076          00521000 T  0044
"TOO MANY","  FILES  ","       ","       ","       ","        ", %077          00522000 T  0044
"FORMAT O","R NAMELI","ST TOO L","ONG     ","       ","      ", %078          00523000 T  0044
"FORMAL P","ARAMETER","  MUST BE","  ARRAY I","DENTIFIE","R     ", %079          00524000 T  0044
0;                                                                               00525000 T  0044
                                                                          12 IS   97 LONG, NEXT SEG     7
        FILL MESSAGE[5,*] WITH                                                   00526000 T  0044
"FORMAL P","ARAMETER","  MUST BE","  SIMPLE ","VARIABLE","      ", %080          00527000 T  0045
                                                                        START OF SEGMENT **********   13
"FORMAL P","ARAMETER","  MUST BE","  FUNCTIO","N IDENTI","FIER    ", %081          00528000 T  0046
"FORMAL P","ARAMETER","  MUST BE","  SUBROUT","INE IDEN","TIFIER  ", %082          00529000 T  0046
"FORMAL P","ARAMETER","  MUST BE","  FUNCTIO","N OR SUB","ROUTINE ", %083          00530000 T  0046
"DO OR IM","PLIED DO","  INDEX M","UST BE I","NTEGER O","R REAL  ", %084          00531000 T  0046
"ILLEGAL ","COMPLEX ","CONSTANT","       ","       ","        ", %085          00532000 T  0046
"ILLEGAL ","MIXED TY","PE STORE","       ","       ","        ", %086          00533000 T  0046
"CONSTANT","  EXCEEDS","  HARDWAR","E LIMITS","       ","      ", %087          00534000 T  0046
```

```
"PARAMETE","R TYPE C","ONFLICTS"," WITH PR","EVIOUS U","SE      ", %088          00535000 T  0046
"COMPLEX ","EXPRESSI","ON ILLEG","AL IN IF","  STATEME","NT      ", %089          00536000 T  0046
"COMPLEX ","EXPRESSI","ON ILLEG","AL IN RE","LATION  ","        ", %090          00537000 T  0046
"TOO MANY","  FORMATS","  REFERN","CED BUT ","NOT YET ","FOUND   ", %091          00538000 T  0046
"VARIABLE","  ARRAY B","OUND MUS","T BE FOR","MAL VARI","ABLE    ", %092          00539000 T  0046
"ARRAY BO","UND MUST","  HAVE IN","TEGER OR","  REAL TY","PE      ", %093          00540000 T  0046
"COMMA OR","  RIGHT P","ARENTHES","IS EXPEC","TED     ","        ", %094          00541000 T  0046
"ARRAY AL","READY GI","VEN BOUN","DS      ","        ","        ", %095          00542000 T  0046
0;                                                                               00543000 T  0046
                                                                          13 IS   97 LONG, NEXT SEG    7
        FILL MESSAGE[6,*] WITH                                                   00544000 T  0047
"ONLY FOR","MAL ARRA","YS MAY B","E GIVEN ","VARIABLE","  BOUNDS ", %096          00545000 T  0047
                                                                        START OF SEGMENT **********   14
"MISSING ","LEFT PAR","ENTHESIS","  IN IMPL","IED DO  ","        ", %097          00546000 T  0049
"SUBSCRIP","T MUST B","E INTEGE","R OR REA","L       ","        ", %098          00547000 T  0049
"ARRAY SI","ZE CANNO","T EXCEED","  32767 W","ORDS    ","        ", %099          00548000 T  0049
"COMMON O","R EQUIV ","BLOCK CA","NNOT EXC","EED 3276","7 WORDS ", %100          00549000 T  0049
"THIS STA","TEMENT I","LLEGAL I","N LOGICA","L IF    ","        ", %101          00550000 T  0049
"REAL OR ","INTEGER ","TYPE REQ","UIRED   ","        ","        ", %102          00551000 T  0049
"ARRAY BO","UND INFO","RMATION ","REQUIRED","        ","        ", %103          00552000 T  0049
"REPLACEM","ENT OPER","ATOR EXP","ECTED   ","        ","        ", %104          00553000 T  0049
"IDENTIFI","ER EXPEC","TED     ","        ","        ","        ", %105          00554000 T  0049
"LEFT PAR","ENTHESIS","  EXPECTE","D       ","        ","        ", %106          00555000 T  0049
"ILLEGAL ","FORMAL P","ARAMETER","        ","        ","        ", %107          00556000 T  0049
"RIGHT PA","RENTHESI","S EXPECT","ED      ","        ","        ", %108          00557000 T  0049
"STATEMEN","T NUMBER","  EXPECTE","D       ","        ","        ", %109          00558000 T  0049
"SLASH EX","PECTED  ","        ","        ","        ","        ", %110          00559000 T  0049
"ENTRY ST","ATEMENT ","CANNOT B","EGIN PRO","GRAM UNI","T       ", %111          00560000 T  0049
0;                                                                               00561000 T  0049
                                                                          14 IS   97 LONG, NEXT SEG    7
        FILL MESSAGE[7,*] WITH                                                   00562000 T  0049
"ARRAY MU","ST BE DI","MENSIONE","D PRIOR ","TO EQUIV","  STMT   ", %112          00563000 T  0050
                                                                        START OF SEGMENT **********   15
"INTEGER ","CONSTANT","  EXPECTE","D       ","        ","        ", %113          00564000 T  0051
"COMMA EX","PECTED  ","        ","        ","        ","        ", %114          00565000 T  0051
"SLASH OR","  END OF ","STATEMEN","T EXPECT","ED      ","        ", %115          00566000 T  0051
"FORMAT, ","ARRAY OR","  NAMELIS","T EXPECT","ED      ","        ", %116          00567000 T  0051
"END OF S","TATEMENT","  EXPECTE","D       ","        ","        ", %117          00568000 T  0051
"IO STATE","MENT WIT","H NAMELI","ST CANNO","T HAVE I","O LIST  ", %118          00569000 T  0051
"COMMA OR","  END OF ","STATEMEN","T EXPECT","ED      ","        ", %119          00570000 T  0051
"STRING T","OO LONG ","        ","        ","        ","        ", %120          00571000 T  0051
"MISSING ","QUOTE AT","  END OF ","STRING  ","        ","        ", %121          00572000 T  0051
"ILLEGAL ","ARRAY BO","UND      ","        ","        ","        ", %122          00573000 T  0051
"TOO MANY","  HANGING","  BRANCHE","S       ","        ","        ", %123          00574000 T  0051
"TOO MANY","  COMMON ","OR EQUIV","ALENCE E","LEMENTS ","        ", %124          00575000 T  0051
"ASTERISK","  EXPECTE","D       ","        ","        ","        ", %125          00576000 T  0051
"COMMA OR","  SLASH E","XPECTED ","        ","        ","        ", %126          00577000 T  0051
"DATA SET","  TOO LAR","GE      ","        ","        ","        ", %127          00578000 T  0051
0;                                                                               00579000 T  0051
                                                                          15 IS   97 LONG, NEXT SEG    7
        FILL MESSAGE[8,*] WITH                                                   00580000 T  0052
"TOO MANY","  ENTRY S","TATEMENT","S IN THI","S SUBPRO","GRAM    ", %128          00581000 T  0052
                                                                        START OF SEGMENT **********   16
"DECIMAL ","WIDTH EX","CEEDS FI","ELD WIDT","H       ","        ", %129          00582000 T  0054
"UNSPECIF","IED FIEL","D WIDTH ","        ","        ","        ", %130          00583000 T  0054
"UNSPECIF","IED SCAL","E FACTOR","        ","        ","        ", %131          00584000 T  0054
"ILLEGAL ","FORMAT C","HARACTER","        ","        ","        ", %132          00585000 T  0054
```

```
"UNSPECIF","IED DECI","MAL FIEL","D       ","          ","          ", %133          00586000 T  0054
"DECIMAL ","FIELD IL","LEGAL FO","R THIS S","PECIFIER","          ", %134          00587000 T  0054
"ILLEGAL ","LABEL   ","          ","          ","          ","          ", %135          00588000 T  0054
"UNDEFINE","D NAMELI","ST      ","          ","          ","          ", %136          00589000 T  0054
"MULTIPLY"," DEFINED"," ACTION ","LABELS  ","          ","          ", %137          00590000 T  0054
"TOO MANY"," NESTED ","DO STATE","MENTS   ","          ","          ", %138          00591000 T  0054
"STMT FUN","CTION ID"," AND EXP","RESSION ","DISAGREE"," IN TYPE", %139          00592000 T  0054
"ILLEGAL ","USE OF S","TATEMENT"," FUNCTIO","N         ","          ", %140          00593000 T  0054
"UNRECOGN","IZED CON","STRUCT  ","          ","          ","          ", %141          00593001 T  0054
"RETURN, ","STOP OR ","CALL EXI","T REQUIR","ED IN SU","BPROGRAM", %142          00593002 T  0054
"FORMAT N","UMBER US","ED PREVI","OUSLY AS"," LABEL  ","          ", %143          00593003 T  0054
0;                                                                                00594000 T  0054
                                                                          16 IS   97 LONG, NEXT SEG    7
        FILL MESSAGE[9,*] WITH                                                    00595000 T  0054
"LABEL US","ED PREVI","OUSLY AS"," FORMAT ","NUMBER  ","          ", %144          00595001 T  0055
                                                                        START OF SEGMENT **********   17
"NON-STAN","DARD RET","URN REQU","IRES LAB","EL PARAM","ETERS    ", %145          00595002 T  0056
"DOUBLE O","R COMPLE","X REQUIR","ES EVEN ","OFFSET  ","          ", %146          00595003 T  0056
"FORMAL P","ARAMETER"," ILLEGAL"," IN DATA"," STATEME","NT        ", %147          00595004 T  0056
"TOO MANY"," LOCAL V","ARIABLES"," IN SOUR","CE PROGR","AM        ", %148          00596000 T  0056
"A $FREEF","ORM SOUR","CE LINE ","MUST HAV","E < 67 C","OLS       ", %149          00596001 T  0056
"A HOL/ST","RING UND","ER $FREE","FORM MUS","T BE ON ","ONE LINE", %150          00596002 T  0056
"THIS CON","STRUCT I","S ILLEGA","L IN TSS"," FORTRAN","          ", %151          00596003 T  0056
"ILLEGAL ","FILE CAR","D PARAME","TER VALU","E         ","          ", %152          00596004 T  0056
"RETURN I","N MAIN P","ROGRAM N","OT ALLOW","ED        ","          ", %153          00596005 T  0056
"NON-POSI","TIVE SUB","SCRIPTS ","ARE ILLE","GAL       ","          ", %154          00596006 T  0056
"NON-IDEN","TIFIER U","SED FOR ","NAME OF ","LIBRARY ","ROUTINE ", %155          00596007 T  0056
"HYPHEN E","XPECTED ","          ","          ","          ","          ", %156          00596008 T  0056
"SEQUENCE"," NUMBER ","EXPECTED","          ","          ","          ", %157          00596009 T  0056
"TOO MANY"," RECURSI","VE CALLS"," ON LIBR","ARY ROUT","INE       ", %158          00596010 T  0056
"ASTERISK"," NOT ALL","OWED ID ","READ STA","TEMENT  ","          ", %159          00596011 T  0056
0;                                                                                00596500 T  0056
                                                                          17 IS   97 LONG, NEXT SEG    7
        FILL MESSAGE[10,*] WITH                                                   00596501 T  0057
"ASTERISK"," ONLY AL","LOWED IN"," FREE FI","ELD OUTP","UT        ", %160          00596510 T  0057
                                                                        START OF SEGMENT **********   18
"TOO MANY"," *-ED LI","ST ELEME","NTS IN O","PUT     ","          ", %161          00596511 T  0059
"HOL OR Q","UOTED ST","RING > 7"," CHARACT","ERS IN E","XPRESSN ", %162          00596512 T  0059
"DECIMAL ","FIELD GR","EATER TH","AN FIELD"," WIDTH-5","          ", %163          00596513 T  0059
"PLUS NOT"," ALLOWED"," IN THIS"," FORMAT ","PHRASE  ","          ", %164          00596514 T  0059
"K NOT AL","LOWED IN"," THIS FO","RMAT PHR","ASE      ","          ", %165          00596515 T  0059
"$ NOT AL","LOWED IN"," THIS FO","RMAT PHR","ASE      ","          ", %166          00596516 T  0059
"INTRNSCS","-NAMD FU","NCT MUST"," HAV PRE","V EXTRNL"," REFERNC", %167          00596517 T  0059
"DATA STM","T COMMON"," ELEM MU","ST BE IN"," BLK DAT","A SUBPRG", %168          00596518 T  0059
"VARIABLE"," CANNOT ","BE A FOR","MAL PARA","METER OR"," IN CMMN", %169          00596519 T  0059
"CURRENT ","SUBPROGR","AM ID EX","PECTED  ","          ","          ", %170          00596520 T  0059
"SUBPROGR","AM, EXTE","RNAL, OR"," ARRAY I","D EXPECT","ED        ", %171          00596521 T  0059
"REPEAT, ","WIDTH, A","ND DECIM","AL PARTS"," MUST BE"," < 4091 ", %172          00596522 T  0059
"REPEAT P","ART MUST"," BE EMPT","Y OR > 0"," AND < 4","091       ", %173          00596523 T  0059
"IN SUBPR","GM:VARBL"," IS USED"," PRIOR T","O USE IN"," DATSTMT", %174          00596524 T  0059
"SYNTATIC","AL TOKEN"," CONTAIN","S TOO MA","NY CHARA","CTERS.  ", %175          00596525 T  0059
0;                                                                                00596526 T  0059
                                                                          18 IS   97 LONG, NEXT SEG    7
        FILL MESSAGE[11,*] WITH                                                   00596527 T  0059
"INTEGER ","VALUE OF"," 8 EXPEC","TED     ","          ","          ", %176          00596528 T  0060
                                                                        START OF SEGMENT **********   19
"INTEGER ","VALUE OF"," 4 EXPEC","TED     ","          ","          ", %177          00596529 T  0061
```

```
"INTEGER ","VALUE OF"," 4 OR 8 ","EXPECTED","        ","        ", %178          00596530 T  0061
"AN ALPHA","BETIC LE","TTER (A,","B,C,...,","Z) IS RE","QUIRED  ", %179          00596531 T  0061
"SECOND R","ANGE LET","TER MUST","  BE GREA","TER THAN","  FIRST  ", %180          00596532 T  0061
"IMPLICIT","  MUST BE","  FIRST S","TATEMENT","  IN PROG","RAM UNIT", %181          00596533 T  0061
"REAL/INT","EGER/LOG","ICAL/COM","PLEX/DOU","BLEPRECI","SION REQ", %182          00596534 T  0061
 "ILLEGAL ","USE OF A","STERISK ","FOR RUN-","TIME EDI","TING    ",%111-          00596535 C  0061
 "NO PROGR","AM UNIT ","FOR THIS","  END STA","TEMENT  ","        ",%112-          00596536 C  0061
0;                                                                               00596599 T  0061
                                                                        19 IS   55 LONG, NEXT SEG    7
END FILL STATEMENTS;                                                             00597000 T  0062
                                                                      START OF SEGMENT **********   20
                                                                        20 IS   13 LONG, NEXT SEG    7
    END ;                                                                        00597950 T  0062
IF DCINPUT THEN                                                                  00598000 T  0062
    BEGIN                                                                        00598020 T  0063
    DCPLACE(ERRORBUFF,N,XTA,LASTSEQ,MESSAGE[N DIV 16,6*(N MOD 16)],              00598040 T  0063
            IF TSSMESTOG THEN TSSMES(TSSMESA[(N-1)DIV 48],                       00598050 T  0068
            ENTIER((N-1) MOD 48)) ELSE FALSE) ;                                  00598060 T  0071
    WRITE(REMOTE,9,ERRORBUFF[*]) ;                                               00598080 T  0075
    END ;                                                                        00598100 T  0079
IF LISTOG OR NOT DCINPUT THEN                                                    00598120 T  0079
    BEGIN                                                                        00598140 T  0081
    PLACE(ERRORBUFF,N,XTA,MESSAGE[N DIV 16,6*(N MOD 16)],ERRORCT,                00598160 T  0082
          LASTERR) ;                                                             00598180 T  0086
    WRITAROW(14,ERRORBUFF) ;                                                     00598200 T  0087
    END ;                                                                        00599000 T  0088
MOVESEQ(LASTERR,LINKLIST) ;                                                      00599500 T  0088
END ERRMESS;                                                                     00600000 T  0090
                                                                         7 IS   92 LONG, NEXT SEG    6
PROCEDURE FLAGROUTINE(NAME1,NAME2,ENTERING) ;                                    00600010 T  0164
PRT(443) = FLAGROUTINE
                VALUE NAME1,NAME2,ENTERING ;                                     00600020 T  0164
                ALPHA NAME1,NAME2 ;                                              00600030 T  0164
                BOOLEAN ENTERING ;                                               00600040 T  0164
                IF ENTERING THEN WRITALIST(FLAGROUTINEFORMAT,7,"ENTERI","NG",NAME1,  00600050 T  0164
                NAME2,"+",FIELD(FLAGROUTINECOUNTER←FLAGROUTINECOUNTER+1),        00600060 T  0167
                FLAGROUTINECOUNTER,0) ELSE                                       00600065 T  0170
                   BEGIN                                                         00600070 T  0170
                   WRITALIST(FLAGROUTINEFORMAT,7,"LEAVIN","G ",NAME1,NAME2,"-",  00600080 T  0174
                   FIELD(FLAGROUTINECOUNTER),FLAGROUTINECOUNTER,0) ;              00600090 T  0176
                   FLAGROUTINECOUNTER←FLAGROUTINECOUNTER-1 ;                     00600100 T  0178
                   END ;                                                         00600110 T  0179
                %END OF FLAGROUTINE                                              00600120 T  0182
                PROCEDURE FLAG(N); VALUE N; INTEGER N;                           00601000 T  0182
PRT(444) = FLAG
                IF NOT ERRORTOG OR DEBUGTOG THEN                                 00602000 T  0182
                BEGIN                                                            00603000 T  0183
                   IF NOT (LISTOG OR SEQERRORS OR DCINPUT) THEN PRINTCARD ;      00604000 T  0183
                   ERRMESS(N);                                                   00606000 T  0187
                   IF ERRORCT ≥ LIMIT THEN GO TO POSTWRAPUP;                     00606500 T  0188
PRT(445) = POSTWRAPUP
PRT(446) = GO TO SOLVER
                END;                                                             00607000 T  0191
                PROCEDURE FLOG(N); VALUE N; INTEGER N;                           00608000 T  0192
PRT(447) = FLOG
                IF NOT ERRORTOG OR DEBUGTOG THEN                                 00609000 T  0192
                   BEGIN ERRORTOG ← TRUE;                                        00610000 T  0193
```

```
                    IF NOT (LISTOG OR SEQERRORS OR DOINPUT) THEN PRINTCARD ;                                  00611000 T  0194
                    ERRMESS(N);                                                                               00613000 T  0198
                    IF ERRORCT ≥ LIMIT THEN GO TO POSTWRAPUP;                                                 00613500 T  0199
                    END;                                                                                      00614000 T  0202
               PROCEDURE FATAL(N);  VALUE N;  INTEGER N ;                                                     00615000 T  0202
PRT(450) = FATAL
               BEGIN                                                                                          00616000 T  0202
                    FORMAT FATALERR("XXXXXXXX", X18,                                                          00617000 T  0202
                                                                                          START OF SEGMENT **********   21
                                                                                          START OF SEGMENT **********   22
PRT(451) = FATALERR
                    "PREVIOUS ERROR IS FATAL - REMAINDER OF SUBPROGRAM IGNORED",                              00618000 T  0000
                    X17, "XXXXXXXX");                                                                         00619000 T  0000
                                                                                          22 IS   19 LONG, NEXT SEG   21
                    ERRORTOG ← FALSE;                                                                         00620000 T  0000
                    FLAG(N);                                                                                  00621000 T  0000
                    WRITALIST(FATALERR,0,0,0,0,0,0,0,0,0) ;                                                   00622000 T  0001
                    WHILE NEXT ≠ 11 DO            % LOOK FOR END STMT                                         00623000 T  0005
               BEGIN                                                                                          00624000 T  0006
                    WHILE NEXT ≠ SEMI DO SCAN;                                                                00625000 T  0006
                    EOSTOG ← TRUE;                                                                            00626000 T  0009
                    SCAN;                                                                                     00627000 T  0010
               END;                                                                                           00628000 T  0010
                    SEGMENT((ADR+4) DIV 4, NSEG, FALSE, EDOC);                                                00629000 T  0011
                    SCAN;                                                                                     00630000 T  0014
               END FATAL;                                                                                     00631000 T  0014
                                                                                          21 IS   15 LONG, NEXT SEG    6
               REAL STREAM PROCEDURE D2B(BCL); BEGIN SI←BCL;  DI←LOC D2B;  DS←80CT END;                        00631100 T  0202
PRT(452) = D2B
               REAL PROCEDURE GET(I);  VALUE I;  INTEGER I ;                                                  00632000 T  0203
PRT(453) = GET
                    BEGIN                                                                                     00633000 T  0204
                         GET ← INFO[(I←I),IR,I,IC] ;                                                          00634000 T  0204
                    END GET;                                                                                  00635000 T  0208
               PROCEDURE PUT(I,VLU);  VALUE I,VLU;  INTEGER I;  REAL VLU ;                                    00636000 T  0210
PRT(454) = PUT
                    BEGIN                                                                                     00637000 T  0210
                         INFO[(I←I),IR,I,IC] ← VLU ;                                                          00638000 T  0210
                    END PUT;                                                                                  00639000 T  0214
               PROCEDURE GETALL(I,INFA,INFB,INFC);                                                            00640000 T  0214
PRT(455) = GETALL
                    VALUE I;  INTEGER I;  REAL INFA,INFB,INFC ;                                               00641000 T  0214
                    BEGIN                                                                                     00642000 T  0214
                         INFA ← INFO[(I←I),IR,I,IC] ;                                                         00643000 T  0214
                         INFB ← INFO[(I←I+1),IR,I,IC];                                                        00644000 T  0218
                         INFC ← INFO[(I←I+1),IR,I,IC];                                                        00645000 T  0222
                    END;                                                                                      00646000 T  0226
               PROCEDURE PUTC(I,V);  VALUE I,V;  INTEGER I;  REAL V ;                                         00646100 T  0226
PRT(456) = PUTC
               IF I←I>SUPERMAXCOM THEN FATAL(124) ELSE COM[I,IR,I,IC]←V ;                                     00646200 T  0226
               REAL PROCEDURE GETC(I);  VALUE I;  INTEGER I ;                                                 00646300 T  0233
PRT(457) = GETC
               GETC←COM[(I←I),IR,I,IC] ;                                                                      00646400 T  0233
               PROCEDURE BAPC(V);  VALUE V;  REAL V ;                                                         00646500 T  0239
PRT(460) = BAPC
               IF NEXTCOM←NEXTCOM+1>SUPERMAXCOM THEN FATAL(124)                                               00646600 T  0239
               ELSE COM[NEXTCOM,IR,NEXTCOM,IC]←V ;                                                            00646700 T  0242
```

```
                    STREAM PROCEDURE MOVEW(F,T,D,M); VALUE D,M;                                    00647000 T  0246
PRT(461) = MOVEW
                        BEGIN SI ← F; DI ← T; D(DS ←32 WDS; DS ← 32 WDS)); DS ← M WDS; END;        00648000 T  0246
                    PROCEDURE CALLEQUIV(I,B,G); VALUE I,B,G; INTEGER I; REAL G; BOOLEAN B ;        00649000 T  0250
PRT(462) = CALLEQUIV
                    BEGIN REAL A,T,R,INFA,INFC;                                                    00650000 T  0250
                                                                                  START OF SEGMENT **********   23
STACK(F+2) = A
STACK(F+3) = T
STACK(F+4) = R
STACK(F+5) = INFA
STACK(F+6) = INFC
                      REAL LAST,D; LABEL L;                                                        00650100 T  0000
STACK(F+7) = LAST
STACK(F+10) = D
                    PROCEDURE CORRECTINFO(R,A); VALUE R,A; INTEGER R,A;                            00651000 T  0000
PRT(463) = CORRECTINFO
                    COMMENT  THIS PROCEDURE CORRECTS THE INFO TABLE FOR COMMON AND                 00651010 T  0000
                               EQUIVALENCE ELEMENTS.                                               00651020 T  0000
                             IF ITEMS ARE IN COMMON THE INFA[EQ] BIT IS SET                        00651030 T  0000
                               THIS BIT IS USED TO TELL DATA STATEMENTS THAT THIS IS A              00651040 T  0000
                               COMMON ELEMENT AND IF NOT IN BLOCK DATA SUBPROGRAM EMIT              00651050 T  0000
                               SYNTAX ERROR 168,                                                   00651060 T  0000
                             THE INFA[CE] BIT IS SET TO TELL THAT IF THE CLASS IS VARID             00651070 T  0000
                               EMIT CODE AS IF THIS ITEM WERE ARRAYID BUT DO NOT                   00651080 T  0000
                               ALLOW SUBSCRIPTING;                                                 00651090 T  0000
                    BEGIN                                                                          00652000 T  0000
                       REAL T,I,LAST,L,REL,TWODBIT,INFA,INFB,INFC;                                 00653000 T  0000
                                                                                  START OF SEGMENT **********   24
STACK(F+2) = T
STACK(F+3) = I
STACK(F+4) = LAST
STACK(F+5) = L
STACK(F+6) = REL
STACK(F+7) = TWODBIT
STACK(F+10) = INFA
STACK(F+11) = INFB
STACK(F+12) = INFC
                      REAL CEBIT;                                                                  00654000 T  0000
STACK(F+13) = CEBIT
                      DEFINE LB = LOWERBOUND#;                                                     00655000 T  0000
                      TWODBIT ← REAL(LENGTH > 1023);                                               00656000 T  0000
                      CEBIT ← REAL(LENGTH > 1);                                                    00657000 T  0001
                      T ← R;                                                                       00658000 T  0002
                      DO                                                                           00659000 T  0003
                      BEGIN                                                                        00660000 T  0004
                         LAST←GETC(T).LASTC-1 ;                                                    00661000 T  0004
                         FOR I ← T+2 STEP 1 UNTIL LAST DO                                          00662000 T  0006
                         BEGIN                                                                     00663000 T  0010
                            IF GETC(I).CLASS = ENDCOM THEN I←GETC(I).LINK ;                        00664000 T  0010
                            INFA←GET(L←GETC(I).LINK); INFC←GET(L+2) ;                              00665000 T  0014
                            IF INFC    > 0 AND LB ≠ 0 THEN                                         00666000 T  0019
                            BEGIN                                                                  00667000 T  0020
                               IF BOOLEAN(INFA    .ADJ) THEN                                       00668000 T  0021
                                  REL ← -INFC.BASE ELSE REL ← INFC.BASE;                           00669000 T  0022
                               PUT(L+2,-(INFC&(REL-LB)[TOBASE]));                                  00670000 T  0025
                            END;                                                                   00671000 T  0028
```

```
        PUT(L,INFA&TWODBIT[TOADJ]&CEBIT[TOCE]&A[TOEQ]);                                  00672000 T  0028
      END;                                                                               00673000 T  0032
   END UNTIL T←GETC(T).ADDR=R ;                                                          00674000 T  0033
END CORRECTINFO;                                                                         00675000 T  0035
                                                                                  24 IS   40 LONG, NEXT SEG   23
   LABEL XIT;                                                                            00676000 T  0000
IF DEBUGTOG THEN FLAGROUTINE(" CALLE","QUIV  ",TRUE ) ;                                  00676010 T  0000
COMMENT  THIS PROCEDURE MAKES THE DETERMINATION IF THIS GROUP                            00676100 T  0002
           IS EQUIVALENCE OR COMMON IF BOTH TREATED AS COMMON;                           00676110 T  0002
   T ← 1;                                                                                00677000 T  0002
      GROUPPRT ← LENGTH ← A ← 0;                                                         00678000 T  0002
      LOWERBOUND ← 32000;                                                                00679000 T  0004
      DO BEGIN IF GETC(T).CE=1 THEN                                                      00680000 T  0005
      BEGIN IF GROUPPRT ≠ 0 THEN                                                         00681000 T  0007
         BEGIN XTA←G ;                                                                   00682000 T  0009
           FLAG(2);                                                                      00683000 T  0010
         END;                                                                            00684000 T  0011
         GROUPPRT←GET(A←GETC(T+1)).ADDR ;                                                00685000 T  0011
      END;                                                                               00686000 T  0014
      IF T > R THEN R ← T;                                                               00687000 T  0014
      END UNTIL T←GETC(T).ADDR=1 ;                                                       00688000 T  0016
      IF GROUPPRT = 0 THEN                                                               00689000 T  0019
         IF B THEN BEGIN BUMPPRT;GROUPPRT←PRTS END ELSE  %OWN                            00689100 T  0019
         BEGIN BUMPLOCALS; GROUPPRT←LOCALS + 1536 END;                                   00690000 T  0025
   T ← R;                                                                                00691000 T  0034
   SWARYCT ← 0;                                                                          00691500 T  0034
   SSNM[6]←LOCALS  ;   SSNM[7]←PARMS   ;   SSNM[8]←PRTS    ;                            00691510 T  0035
   SSNM[9]←LSTS    ;  SSNM[10]←LSTA    ;  SSNM[11]←TV      ;                            00691520 T  0039
  SSNM[12]←SAVESUBS;  SSNM[13]←NAMEIND ;  SSNM[14]←LSTI    ;                            00691530 T  0043
  SSNM[15]←FX1     ;  SSNM[16]←FX2     ;  SSNM[17]←FX3     ;                            00691540 T  0046
 SSNM[18]←NX1     ;                                                                      00691550 T  0050
  SEENADOUB←FALSE;%                                                                     00691800 T  0051
  DO IF GETC(T+1) ≠ 0 THEN BEGIN EQUIV(T); T←R; END                                     00691900 T  0053
      ELSE T←GETC(T).ADDR UNTIL T = R;                                                  00691950 T  0057
  IF GETC(T) > 0 THEN EQUIV(T);                                                         00692000 T  0063
  LOCALS←SSNM[6];  PARMS←SSNM[7];  PRTS←SSNM[8];  LSTS←SSNM[9] ;                        00692010 T  0065
  LSTA←SSNM[10];  TV←SSNM[11];  SAVESUBS←SSNM[12];  NAMEIND←SSNM[13] ;                  00692020 T  0069
  LSTI←SSNM[14];  FX1←SSNM[15];  FX2←SSNM[16];  FX3←SSNM[17] ;                          00692030 T  0073
  NX1←SSNM[18] ;                                                                        00692040 T  0077
  LENGTH ← LENGTH + LOWERBOUND;                                                         00693000 T  0078
  IF SEENADOUB THEN    LENGTH←LENGTH+LENGTH.[47:1];% EVEN UP LENGTH                     00693500 T  0079
  SEENADOUB←FALSE;                                                                      00693600 T  0082
  IF A = 0 THEN                                                                         00694000 T  0084
  BEGIN COMMENT THIS IS AN EQUIVALENCE GROUP;                                           00694100 T  0085
  IF SWARYCT ≥ 1 AND LENGTH = 1 THEN LENGTH ← 2;                                        00694150 T  0085
     IF LENGTH > 1 THEN                                                                 00694200 T  0088
     BEGIN                                                                              00694300 T  0089
        A ← ENTER(0 & ARRAYID[TOCLASS] & GROUPPRT[TOADDR],                              00695000 T  0090
           EQVID ← EQVID+1);                                                            00696000 T  0092
        PUT(A+2, 0 & LENGTH[TOSIZE]);                                                   00696100 T  0094
     END;                                                                               00696200 T  0096
     CORRECTINFO(R,0);                                                                  00697100 T  0096
     GO TO XIT;                                                                         00697200 T  0097
  END;                                                                                  00697300 T  0098
  COMMENT THIS IS A COMMON BLOCK;                                                       00697400 T  0098
  IF T ← (INFC ← GET(A+2)).SIZE ≠ 0 THEN                                                00700000 T  0098
  IF T ≤ 1023 AND LENGTH > 1023 THEN                                                    00701000 T  0101
```

```
                    BEGIN XTA ← GET(A+1); FLAG(47) END;                                          00702000 T  0103
                 IF T < LENGTH THEN PUT(A+2, INFC&LENGTH[TOSIZE]) ELSE LENGTH ← T;               00702100 T  0106
                 IF LENGTH = 1 THEN LENGTH ← 2;                                                  00703000 T  0111
                 CORRECTINFO(R,1);                                                               00704000 T  0113
                 XIT:                                                                            00705000 T  0114
                 IF LENGTH > 32767 THEN BEGIN XTA ← GET(A+1); FLAG(100) END;                     00705100 T  0115
              IF DEBUGTOG THEN FLAGROUTINE(" CALLE","QUIV  ",FALSE) ;                            00705110 T  0118
              END CALLEQUIV;                                                                     00706000 T  0120
                                                                                    23 IS  127 LONG, NEXT SEG    6
              STREAM PROCEDURE STOSEQ(XR,SEQ,ID);  VALUE ID;                                     00706010 T  0250
PRT(464) = STOSEQ
              BEGIN LOCAL T;  LABEL L1,L2 ;                                                      00706015 T  0250
                    SI ← LOC ID; DI←XR; DS←7LIT"0"; DS←CHR; SI←SI+1;                             00706020 T  0250
                    IF SC ≥ "0" THEN                                                             00706025 T  0252
                    BEGIN SI←SI+4; DI←DI-2;                                                      00706030 T  0252
                          5(IF SC= " " THEN SI ← SI-1                                            00706035 T  0253
                            ELSE BEGIN DS←CHR; SI←SI-2; DI←DI-2; END);                           00706040 T  0254
                    END ELSE                                                                     00706045 T  0256
                    BEGIN DI←DI-6;                                                               00706050 T  0256
                          6(IF SC=" " THEN BEGIN TALLY←TALLY+1; SI←SI+1 END                      00706055 T  0256
                            ELSE DS←CHR);                                                        00706060 T  0258
                          T←TALLY;                                                               00706065 T  0259
                    END;                                                                         00706070 T  0259
                    DI←XR; DI←DI+8; SI←SEQ;                                                      00706075 T  0259
                    8(IF SC=" " THEN SI←SI+1 ELSE JUMP OUT TO L1);  DS←8LIT"BLANKSEQ";           00706080 T  0260
                    GO L2;  L1:   SI←SEQ;  DS←WDS;  L2:                                          00706085 T  0263
                    DI←DI+4; SI←LOC T; SI←SI+7; DS←CHR;                                          00706090 T  0264
              END;                                                                               00706095 T  0266
              PROCEDURE ENTERX(IDENT,ADINFO); VALUE IDENT, ADINFO;                               00706120 T  0266
PRT(465) = ENTERX
              REAL IDENT,ADINFO;                                                                 00706150 T  0266
              BEGIN REAL R,S ;                                                                   00706200 T  0266
                                                                                  START OF SEGMENT **********   25
STACK(F+2) = R
STACK(F+3) = S
                    IF ADINFO.CLASS>LABELID THEN                                                 00706220 T  0000
                       BEGIN                                                                     00706240 T  0001
                       XR[2]←ADINFO;  STOSEQ(XR,LINKLIST,IDENT) ;                                00706260 T  0001
                       IF ADINFO.CLASS=FUNID THEN XR[2].[26:1]←S←1 ;                             00706280 T  0005
                       WRITE(XREFG,3,XR[*]);  XGLOBALS←TRUE ;                                    00706300 T  0009
                       END ;                                                                     00706320 T  0015
                    IF R←XRI MOD XRBUFFDIV3=0 THEN                                               00706340 T  0015
                       IF XRI≠0 THEN WRITE(XREFF,XRBUFF,XRRY[*]) ;                               00706350 T  0017
                    XRRY[(R←R+R+R)+2]←ADINFO&S[26:47:1] ;                                        00706360 T  0023
                    STOSEQ(XRRY[R],LINKLIST,IDENT) ;                                             00706380 T  0027
                    IF XRI←XRI+1=1 THEN BEGIN XR[3]←IDENT;  XR[4]←XRRY[2] END ;                  00706450 T  0029
              END ENTERX;                                                                        00706500 T  0034
                                                                                    25 IS   37 LONG, NEXT SEG    6
                 STREAM PROCEDURE TRANSFER(W2,B,K);  VALUE K ;                                   00706525 T  0266
PRT(466) = TRANSFER
                    BEGIN DI←W2; SI←B; SI←SI+8; K(DS←WDS; SI←SI+16) END ;                        00706530 T  0266
              BOOLEAN STREAM PROCEDURE CMPA(F,S);                                                00706560 T  0269
PRT(467) = CMPA
              BEGIN SI← F; DI ← S; IF 8 SC ≤ DC THEN TALLY ← 1; CMPA←TALLY; END;                  00706570 T  0269
              BOOLEAN PROCEDURE CMP(A,B);  ARRAY A,B[0];                                         00706600 T  0272
PRT(470) = CMP
              BEGIN                                                                              00706610 T  0272
```

```
                CMP ← IF A[0] < B[0] THEN TRUE ELSE IF A[0] = B[0] THEN                                  00706640 T  0272
                      IF A[2].[26:4] < B[2].[26:4] THEN TRUE ELSE                                        00706650 T  0276
                      IF A[2].[26:4] = B[2].[26:4] THEN CMPA(A[1],B[1]) ELSE FALSE                       00706660 T  0280
                      ELSE FALSE;                                                                        00706670 T  0286
                END;                                                                                     00706680 T  0287
                PROCEDURE HV(V); ARRAY V[0];                                                             00706700 T  0289
PRT(471) = HV
                FILL V[*] WITH OCT777777777777777,OCT777777777777777,OCT777777777777777;                 00706720 T  0289
                                                                                                  START OF SEGMENT **********   26
                BOOLEAN PROCEDURE INP(XR); ARRAY XR[0];                                                  00706730 T  0291
                                                                                                   26 IS    3 LONG, NEXT SEG    6
PRT(472) = INP
                BEGIN LABEL EOF; REAL R ;                                                                00706733 T  0292
                                                                                                  START OF SEGMENT **********   27
STACK(F+3) = R
                IF EODS THEN READ(XREFG,3,XR[*])[EOF] ELSE                                               00706735 T  0000
PRT(473) = EOF
                IF IT ← IT+1 > XRI THEN                                                                  00706750 T  0006
                     BEGIN EOF; INP←TRUE; IT←XRI←0; END                                                  00706760 T  0008
                ELSE BEGIN                                                                               00706770 T  0011
                    IF R←(IT-1) MOD XRBUFFDIV3=0 THEN READ(XREFF,XRBUFF,XRRY[*]) ;                       00706780 T  0011
                    XR[0]←XRRY[R←R+R+R]; XR[2]←XRRY[R+2]; TRANSFER(XR[1],XRRY[R],1) ;                     00706800 T  0018
                    END ;                                                                                00706810 T  0025
                END OF INP ;                                                                             00706820 T  0025
                                                                                                   27 IS   30 LONG, NEXT SEG    6
                PROCEDURE VARIABLEDIMS(A,C); VALUE A, C; REAL A, C;                                      00707000 T  0292
PRT(474) = VARIABLEDIMS
                BEGIN                                                                                    00708000 T  0292
                   REAL NSUBS, BDLINK, BOUND, I;                                                         00709000 T  0292
                                                                                                  START OF SEGMENT **********   28
STACK(F+2) = NSUBS
STACK(F+3) = BDLINK
STACK(F+4) = BOUND
STACK(F+5) = I
                   LABEL XIT;                                                                            00710000 T  0000
                IF DEBUGTOG THEN FLAGROUTINE("VARIAB","LEDIMS",TRUE ) ;                                  00710010 T  0000
                   IF NSUBS ← C.NEXTRA = 1 AND A.SUBCLASS < DOUBTYPE THEN GO TO XIT;                     00711000 T  0002
                   BDLINK ←    C.ADINFO;                                                                 00712000 T  0006
                   FOR I ← 1 STEP 1 UNTIL NSUBS DO                                                       00713000 T  0007
                   BEGIN                                                                                 00714000 T  0010
                      IF BOUND ← EXTRAINFO[BDLINK.IR,BDLINK.IC] < 0 THEN                                 00715000 T  0010
                         EMITOPDCLIT(BOUND) ELSE EMITNUM(BOUND);                                         00716000 T  0013
                      BDLINK ← BDLINK-1;                                                                 00717000 T  0015
                      IF I > 1 THEN EMITO(MUL);                                                          00718000 T  0017
                   END;                                                                                  00719000 T  0019
                   IF A.SUBCLASS ≥ DOUBTYPE THEN EMITPAIR(2, MUL);                                       00720000 T  0021
                   EMITPAIR(   C.SIZE, STD);                                                             00721000 T  0024
                   XIT:                                                                                  00722000 T  0025
                IF DEBUGTOG THEN FLAGROUTINE("VARIAB","LEDIMS",FALSE) ;                                  00722010 T  0026
                END VARIABLEDIMS;                                                                        00723000 T  0028
                                                                                                   28 IS   33 LONG, NEXT SEG    6
                PROCEDURE EMITD(R,OP); VALUE R,OP; REAL R,OP; FORWARD;                                   00723100 T  0292
PRT(475) = EMITD
                STREAM PROCEDURE SETPNT(W1,W2,PNT,CL,SUBC,STAR,POS,F,S,Q) ;                              00723140 T  0292
PRT(476) = SETPNT
                VALUE CL,SUBC,STAR,POS,F,S,Q ;                                                           00723150 T  0292
                BEGIN F(SI←W1; SI←SI+2; DI←PNT ;                                                         00723160 T  0292
```

```
                    IF SC≥"0" THEN                                                          00723170 T  0293
                    BEGIN DS←5CHR; DS←LIT" "; DI←DI-6; DS←5FILL ;                           00723180 T  0294
                    END ELSE BEGIN DS←6LIT" "; DI←PNT; DS←Q CHR END) ;                      00723200 T  0296
                   S(DI←PNT; DI←DI+6; DS←LIT" ";                                            00723210 T  0298
                  S1 ← LOC SUBC; SI ← SI+2; DS ← 6 CHR; DS ← LIT " ";                       00723230 T  0300
                    SI←LOC CL; SI←SI+2; DS←6CHR; DS←LIT" ");                                00723240 T  0301
                   DI←PNT; DI←DI+21;                                                        00723250 T  0303
                    POS(DI←DI+11);                                                          00723255 T  0303
                  S1 ← LOC STAR; SI ← SI+7; DS ← CHR;                                       00723265 T  0305
                  S1 ← W2;  DS←8CHR ;                                                       00723280 T  0305
                  S1 ← LOC STAR; SI ← SI+7; DS ← CHR;                                       00723295 T  0306
              END OF SETPNT ;                                                               00723325 T  0307
              PROCEDURE PT(EOF,REC);VALUE EOF; BOOLEAN EOF; ARRAY REC[0];                   00723450 T  0307
PRT(477) = PT
              BEGIN  LABEL L6;  REAL L ;                                                    00723470 T  0307
                                                                                  START OF SEGMENT **********   29
STACK(F+2) = L
                 IF EOF THEN WRITE(LINE,15,PRINTBUFF[*])                                    00723520 T  0000
                 ELSE                                                                       00723595 T  0003
                  IF BOOLEAN(L←REAL(REC[0]←REC[0]&REC[2] [1:26:1]=XTA)) THEN                00723610 T  0005
                       BEGIN                                                                00723620 T  0009
                       IF IT←IT+1=9 THEN                                                    00723630 T  0009
                            BEGIN LASTLINE←FALSE; WRITE(LINE,15,PRINTBUFF[*]) ;             00723640 T  0011
              L6:           BLANKIT(PRINTBUFF,15,IT←0) ;                                    00723650 T  0018
                            END ;                                                           00723660 T  0020
                       SETPNT(REC[0],REC[1],PRINTBUFF,KLASS[REC[2],CLASS],TYPES[REC[2]      00723670 T  0020
                             ,SUBCLASS],IF BOOLEAN(REC[2]) THEN "*" ELSE " ",IT,L-1,        00723680 T  0023
                              LASTLINE,6-REC[2],[27:3]) ;                                   00723690 T  0027
                            END                                                             00723700 T  0029
                       ELSE BEGIN                                                           00723710 T  0029
                            LASTLINE←TRUE;  WRITE(RITE,15,PRINTBUFF[*]);  XTA←REC[0] ;      00723720 T  0030
                            GO L6 ;                                                         00723730 T  0036
                            END ;                                                           00723740 T  0037
                  END OF PT ;                                                               00723800 T  0037
                                                                                  29 IS   40 LONG, NEXT SEG    6
                  PROCEDURE CHECKINFO;                                                      00724000 T  0307
PRT(500) = CHECKINFO
                  BEGIN REAL I, T, A;                                                       00725000 T  0307
                                                                                  START OF SEGMENT **********   30
STACK(F+2) = I
STACK(F+3) = T
STACK(F+4) = A
                       REAL LASTF;                                                          00725100 T  0000
STACK(F+5) = LASTF
                       REAL INFB,INFC;                                                      00726000 T  0000
STACK(F+6) = INFB
STACK(F+7) = INFC
                       LABEL NXT; ALPHA N;                                                  00727000 T  0000
STACK(F+10) = N
                       FORMAT INFOF( 3(A6, X2),                                             00728000 T  0000
                                                                                  START OF SEGMENT **********   31
PRT(501) = INFOF
                            " ADDRESS = ", A2, A4,                                          00729000 T  0000
                            ", LENGTH = ", I5,                                              00730000 T  0000
                            ", OFFSET = ", I5),                                             00731000 T  0000
                          INFOT( / "LOCAL IDENTIFIERS:"),                                   00732000 T  0000
                          LABF(A6,X10,"LABEL    REL-ADR = ",I6,", SEGMNT = ",I5);         00733000 T  0000
```

```
IF PRTOG THEN                                                          00734000 T  0000
WRITALIST(INFOT,0,0,0,0,0,0,0,0,0) ;                                   00735000 T  0000
IF I ← SEARCH("ERR   ") ≠ 0 THEN                                       00736000 T  0004
   IF GET(I).CLASS = UNKNOWN THEN PUT(I+1, ".....");                   00737000 T  0006
IF I ← SEARCH("END   ") ≠ 0 THEN                                       00738000 T  0010
   IF GET(I).CLASS = UNKNOWN THEN PUT(I+1, ".....");                   00739000 T  0012
IF NOT DCINPUT THEN IF I←SEARCH("ZIP   ") ≠ 0 THEN                     00739100 T  0016
   IF GET(I).CLASS=UNKNOWN THEN PUT(I+1,".....") ;                     00739200 T  0020
FOR I ← 0 STEP 1 UNTIL NEXTCOM DO                                      00740000 T  0024
IF GETC(I).CLASS=HEADER THEN                                           00741000 T  0029
                                                                       00742000 T  0030
IF GETC(I).CE=1 THEN                                                   00743000 T  0030
PUT((T←GETC(I+1).LINK+2),GET(T)&0[TOADINFO]) ;                         00744000 T  0033
                                                                       00745000 T  0040
T ← 2; WHILE T < NEXTINFO DO                                           00746000 T  0040
BEGIN                                                                  00747000 T  0043
   GETALL(T,A,INFB,INFC);                                              00748000 T  0043
   IF I ← A.CLASS > FILEID THEN GO TO NXT;                             00749000 T  0044
   IF N ← INFB    = "....." THEN GO TO NXT;                            00750000 T  0047
   XTA ← N;                                                            00751000 T  0048
   IF I = LABELID THEN                                                 00752000 T  0049
   BEGIN                                                               00753000 T  0050
      IF A > 0 THEN FLAG(19) ELSE                                      00754000 T  0050
      IF PRTOG THEN WRITALIST(LABF,3,N,A.ADDR DIV 4,A,SEGNO,0,0,0,0,   00755000 T  0052
      0) ;                                                             00755010 T  0060
      GO TO NXT;                                                       00756000 T  0061
   END;                                                                00757000 T  0061
   IF I = FORMATID THEN                                                00758000 T  0061
      IF A > 0 THEN BEGIN FLAG(62); GO TO NXT END;                     00759000 T  0062
IF I = NAMELIST THEN                                                   00760000 T  0065
IF A > 0 THEN BEGIN FLAG(136); GO TO NXT END;                          00761000 T  0066
IF I = ARRAYID THEN                                                    00762000 T  0069
   IF BOOLEAN(A,CE) THEN BEGIN IF A>0 THEN T←GETSPACE(T) END           00763000 T  0070
   ELSE IF A<0 THEN                                                    00764000 T  0074
   IF BOOLEAN(A,FORMAL) THEN                                           00764020 T  0075
   BEGIN IF SPLINK > 1 AND ELX > 1 THEN                                00764040 T  0076
      BEGIN % THIS IS A FORMAL PARAMETER FOR A SUBROUTINE OR A         00764060 T  0079
            % FUNCTION THAT HAS ONE OR MORE ENTRY STATEMENT. THIS      00764070 T  0079
            % PARAMETER WILL BE INITIALIZED TO AN ARRAY DESCRIPTOR     00764080 T  0079
            % TO 0 SO THAT IF THE PARAMETERS ARE NOT SET UP BY THE     00764090 T  0079
            % CALL ANY REFERENCE TO THEM WILL CAUSE AN INVALID         00764100 T  0079
            % ADDRESS                                                  00764110 T  0079
        IF LASTF = 0 THEN % FIRST TIME THRU                            00764150 T  0079
        BEGIN LASTF ← A.ADDR;                                          00764200 T  0080
              EMITDESCLIT(2); EMITL(1);                                00764220 T  0082
              EMITD(50,DIA); EMITD(10,DIB); EMITD(10,TRB);             00764240 T  0083
        END ELSE EMITPAIR(LASTF,LOD);                                  00764260 T  0086
        EMITPAIR(A,ADDR,SND);                                          00764280 T  0088
      END                                                              00764300 T  0089
   END                                                                 00764320 T  0089
   ELSE ARRAYDEC(T);                                                   00765000 T  0089
IF PRTOG THEN                                                          00766000 T  0090
   WRITALIST(INFOF,7,INFB,                                             00767000 T  0091
   IF BOOLEAN(A,TYPEFIXED) THEN TYPES[A,SUBCLASS] ELSE "      ",       00768000 T  0093
   KLASS[A,CLASS],                                                     00769000 T  0096
   IF A.ADDR < 1024 AND A < 0 THEN "R+" ELSE "  ",                     00770000 T  0097
```

```
                        IF A < 0 THEN B2D(A,[26:10]) ELSE "NULL",                                  00771000 T  0101
                        INFC,SIZE,                                                                 00772000 T  0104
                        INFC,BASE,0) ;                                                             00773000 T  0105
                     IF BOOLEAN(A.CE) THEN IF A.SUBCLASS ≥ DOUBTYPE THEN                           00773100 T  0106
                        IF BOOLEAN(INFC,BASE) THEN FLAG(146);                                      00773200 T  0109
                     NXT;                                                                          00774000 T  0111
                        T ← T+3;                                                                   00775000 T  0112
                  END;                                                                             00776000 T  0113
               END CHECKINFO;                                                                      00777000 T  0119
                                                                                  30 IS  123 LONG, NEXT SEG     6
               PROCEDURE SEGMENTSTART;                                                             00778000 T  0307
PRT(502) = SEGMENTSTART
               BEGIN                                                                               00779000 T  0307
                  NSEG ← NXAVIL ← NXAVIL + 1;                                                      00780000 T  0307
                  IF LISTOG THEN WRITALIST(SEGSTRT,1,NSEG,0,0,0,0,0,0,0) ;                         00781000 T  0309
                  DEBUGADR←ADR←-1 ;                                                                00782000 T  0314
                  IF NOT SEGOVFLAG THEN                                                            00783000 T  0316
                  BEGIN                                                                            00784000 T  0317
                     DATAPRT←DATASTRT←DATALINK←DATASKP←                                            00784100 T  0317
                     LABELMOM ← BRANCHX ← FUNVAR ←                                                 00785000 T  0317
                     DT ← SPLINK ← NEXTCOM ← ELX ← 0;                                              00786000 T  0317
                     NEXTSS ← 1022;                                                                00787000 T  0323
                     INITIALSEGNO ← NSEG;                                                          00788000 T  0324
                     FOR I ← 0 STEP 1 UNTIL LBRANCH DO BRANCHES[I] ← I+1;                          00789000 T  0324
                  FOR I ← 0 STEP 1 UNTIL SHX DO STACKHEAD[I] ← 0;                                  00790000 T  0330
                     NEXTINFO ← LOCALS ← 2;                                                        00791000 T  0334
                     F2TOG ← FALSE; RETURNFOUND ← FALSE;                                           00792000 T  0335
                  END;                                                                             00793000 T  0339
                  ENDSEGTOG ← FALSE;                                                               00794000 T  0339
                  IF SEGSW THEN LINESEG[NOLIN←0,0]←0 & D2B(LASTSEQ)[10:20:28] ;                    00794400 T  0341
               END SEGMENTSTART;                                                                   00795000 T  0347
               PROCEDURE FIXPARAMS(I); VALUE I; INTEGER I;                                         00796000 T  0347
PRT(503) = FIXPARAMS
               BEGIN                                                                               00797000 T  0347
                  REAL FMINUS, NPARMS, ELINK, LABX;                                                00798000 T  0347
                                                                                  START OF SEGMENT **********    32
STACK(F+2) = FMINUS
STACK(F+3) = NPARMS
STACK(F+4) = ELINK
STACK(F+5) = LABX
                  REAL PLINKX, PLINK;                                                              00799000 T  0000
STACK(F+6) = PLINKX
STACK(F+7) = PLINK
                  REAL PTYPEX, PTYPE;                                                              00800000 T  0000
STACK(F+10) = PTYPEX
STACK(F+11) = PTYPE
                  REAL CL, INF;                                                                    00801000 T  0000
STACK(F+12) = CL
STACK(F+13) = INF
                  LABEL ARRY, LOAD, INDX;                                                          00802000 T  0000
                  REAL EWORD;                                                                      00803000 T  0000
STACK(F+14) = EWORD
                  IF DEBUGTOG THEN FLAGROUTINE("  FIXP","ARMS  ",TRUE ) ;                         00803010 T  0000
                     EWORD ← ENTRYLINK[I];                                                         00804000 T  0002
                     FLINK ← EWORD.LINK;                                                           00805000 T  0003
                     IF LABX ← EWORD.CLASS > 0 THEN                                                00806000 T  0004
                           BEGIN                                                                   00807000 T  0006
```

```
      EMITO(MKS);                                                                    00808000 T  0006
      EMITDESCLIT(LABELMOM);                                                         00809000 T  0007
      EMITL(LABX);                                                                   00810000 T  0008
      EMITL(1); EMITL(1); EMITL(0);                                                  00811000 T  0008
      EMITOPDCLIT(BLKCNTRLINT);                                                      00812000 T  0011
      EMITL(1); EMITPAIR(FPLUS2,STD);%   F+2←TRUE FOR BLKXIT CALL                    00812400 T  0011
      END;                                                                %106-      00813000 P  0013
FMINUS ← 1920;                                                                       00814000 T  0013
NPARMS ← (J←GET(ELINK+2)),NEXTRA;                                                    00815000 T  0014
 IF NPARMS > 0 THEN                                                     %106-       00816000 P  0017
BEGIN                                                                                00817000 T  0017
   PLINKX ← EWORD.ADDR-NPARMS+1;                                                     00818000 T  0018
   PTYPEX ← J.ADINFO + NPARMS-1;                                                     00819000 T  0020
   FOR J ← 1 STEP 1 UNTIL NPARMS DO                                                  00820000 T  0022
   BEGIN                                                                             00821000 T  0028
      PLINK ← EXTRAINFO[PLINKX.IR,PLINKX.IC];                                        00822000 T  0028
      PTYPE ← EXTRAINFO[PTYPEX.IR,PTYPEX.IC].CLASS;                                  00823000 T  0030
      FMINUS ← FMINUS+1;                                                             00824000 T  0034
      IF PLINK = 0 THEN                                                              00825000 T  0035
      BEGIN                                                                          00826000 T  0036
         EMITOPDCLIT(FMINUS);                                                        00827000 T  0036
         EMITL(LABX ← LABX-1);                                                       00828000 T  0037
         EMITDESCLIT(LABELMOM);                                                      00829000 T  0039
         EMITO(STD);                                                                 00830000 T  0039
      END ELSE                                                                       00831000 T  0040
      BEGIN                                                                          00832000 T  0040
      IF CL ← (INF ←  GET(PLINK)).CLASS = UNKNOWN THEN CL ← VARID;                   00833000 T  0041
      XTA ←  GET(PLINK+1);                                                           00834000 T  0045
      IF PTYPE = 0 THEN EXTRAINFO[PTYPEX.IR,PTYPEX.IC].CLASS                         00835000 T  0046
      ← PTYPE ← CL;                                                                  00836000 T  0050
      CASE PTYPE OF                                                                  00837000 T  0052
      BEGIN ;                                                                        00838000 T  0053
PRT(504) = *CASE STATEMENT DESCRIPTOR*
      IF CL ≠ ARRAYID THEN FLAG(79) ELSE                                             00839000 T  0053
      ARRY;                                                                          00840000 T  0055
      BEGIN                                                                          00841000 T  0056
         FMINUS ← FMINUS+1;                                                          00842000 T  0056
         IF INF < 0 THEN                                                             00843000 T  0057
         BEGIN                                                                       00844000 T  0058
            EMITPAIR(FMINUS, LOD);                                                   00845000 T  0058
            EMITPAIR(T←INF.ADDR, STD);                                               00846000 T  0059
            EMITOPDCLIT(FMINUS-1);                                                   00847000 T  0061
            EMITPAIR(T-1, STD);                                                      00848000 T  0062
         END;                                                                        00849000 T  0064
      END;                                                                           00850000 T  0064
      IF CL ≠ VARID THEN FLAG(80) ELSE                                               00851000 T  0064
      LOAD;                                                                          00852000 T  0066
      BEGIN                                                                          00854000 T  0068
      IF INF.SUBCLASS≥DOUBTYPE AND CL=VARID THEN FMINUS←FMINUS+1;                    00854100 P  0068
      IF INF<0 THEN                                                                  00854200 T  0072
      BEGIN                                                                          00854300 T  0072
         EMITPAIR(FMINUS, LOD);                                                      00855000 T  0073
         EMITPAIR(T←INF.ADDR,STD);                                                   00856000 T  0074
      IF INF.SUBCLASS≥DOUBTYPE AND CL=VARID THEN                         %105-       00856100 P  0076
         BEGIN EMITOPDCLIT(FMINUS-1);                                                00856200 T  0078
               EMITPAIR(T+1,STD);                                                    00856300 T  0080
         END;                                                                        00856400 T  0081
```

```
END ;
END;
; ; ; ;
IF CL = FUNID THEN GO TO LOAD ELSE FLAG(81);




;
IF CL = FUNID OR CL = SUBRID OR CL = EXTID THEN GO TO LOAD
     ELSE FLAG(83);
IF CL = SUBRID THEN GO TO LOAD ELSE FLAG(82);
; ;
BEGIN
   IF CL = ARRAYID THEN
   BEGIN
      EXTRAINFO[PTYPEX,IR,PTYPEX,IC],CLASS ← CL;
      GO TO ARRY;
   END;
   INDX;
   IF CL ≠ VARID THEN FLAG(80);
   FMINUS ← FMINUS+1;
   IF INF < 0 THEN
   BEGIN
      EMITOPDCLIT(FMINUS-1);
      EMITDESCLIT(FMINUS);
      EMITPAIR(INF,ADDR, STD);
   END;
END;
IF CL = VARID THEN GO TO LOAD ELSE FLAG(80);
GO TO INDX;
END CASE STATEMENT;


A ← INF.SUBCLASS;
IF T ← EXTRAINFO[PTYPEX,IR, PTYPEX,IC].SUBCLASS = 0 OR
   T = INTYPE AND A = REALTYPE THEN
   EXTRAINFO[PTYPEX,IR,PTYPEX,IC].SUBCLASS ← A ELSE
   IF T ≠ A THEN
   BEGIN XTA ← GET(PLINK+1); FLAG(88) END;
END;
PLINKX ← PLINKX+1;
PTYPEX ← PTYPEX-1;
END;
PLINKX ← PLINKX-NPARMS;
FOR J ← 1 STEP 1 UNTIL NPARMS DO
BEGIN
   PLINK ← EXTRAINFO[PLINKX,IR,PLINKX,IC];
   IF PLINK ≠ 0 THEN
   IF (A←GET(PLINK)).CLASS = ARRAYID THEN
   IF T←GET(PLINK+2) < 0 THEN VARIABLEDIMS(A, T);
   PLINKX ← PLINKX+1;
END;
END;
EMITR(GET(ELINK+2),BASE & (GET(ELINK),SEGNO)[TOSEGNO], FALSE);
IF DEBUGTOG THEN FLAGROUTINE("  FIXP","ARMS  ",FALSE) ;
```

```
00856500 T  0081
00857000 T  0081
00858000 T  0082
00859000 T  0082
00859100 T  0084
00859200 T  0084
00859300 T  0084
00859400 T  0084
00859500 T  0084
00860000 T  0084
00861000 T  0084
00861100 T  0087
00862000 T  0089
00863000 T  0092
00864000 T  0092
00865000 T  0092
00866000 T  0092
00867000 T  0093
00868000 T  0097
00869000 T  0098
00870000 T  0098
00871000 T  0098
00872000 T  0100
00873000 T  0101
00874000 T  0102
00875000 T  0102
00876000 T  0103
00877000 T  0104
00878000 T  0106
00879000 T  0106
00880000 T  0106
00892000 T  0109
00893000 T  0109
START OF SEGMENT **********    33
  33 IS   17 LONG, NEXT SEG    32
00894000 T  0110
00895000 T  0111
00896000 T  0115
00897000 T  0117
00898000 T  0121
00899000 T  0123
00900000 T  0126
00901000 T  0126
00902000 T  0127
00903000 T  0128
00904000 T  0130
00905000 T  0132
00906000 T  0133
00907000 T  0133
00908000 T  0135
00909000 T  0136
00910000 T  0139
00911000 T  0143
00912000 T  0144
00913000 T  0147
00914000 T  0147
00914010 T  0151
```

```
                    END FIXPARAMS;                                                                  00915000 T  0153
                                                                                                32 IS  160 LONG, NEXT SEG     6
                    PROCEDURE XREFCORESORT ;                                                        00915100 T  0347
PRT(505) = XREFCORESORT
                        BEGIN                                                                       00915120 T  0347
                        REAL F,G,H,I,J,K,L,T,TT,IJ,M,TN ;                                           00915140 T  0347
                                                                                                START OF SEGMENT **********    34
STACK(F+2) = F
STACK(F+3) = G
STACK(F+4) = H
STACK(F+5) = I
STACK(F+6) = J
STACK(F+7) = K
STACK(F+10) = L
STACK(F+11) = T
STACK(F+12) = TT
STACK(F+13) = IJ
STACK(F+14) = M
STACK(F+15) = TN
                        SAVE ARRAY A[0:XRI-1], IL,IU[0:8] ;                                         00915160 T  0000
STACK(F+16) = A
STACK(F+17) = IL
STACK(F+20) = IU
                        ARRAY W2,W3[0:XRI-1] ;                                                      00915180 T  0005
STACK(F+21) = W2
STACK(F+22) = W3
                        LABEL L1,L2,L3,L4,L5,L6,L11,L12,L13,L14 ;                                   00915200 T  0008
                        DEFINE CMPARGT(A) = A.NAME=TN THEN IF (IF G←W3[H←A.[2:10]].[26:4]            00915220 T  0008
                                =TT←W3[F←T.[2:10]].[26:4] THEN NOT CMPA(W2[H],                      00915240 T  0008
                                W2[F]) ELSE G>TT) #,                                                00915260 T  0008
                            CMPARLS(A) = A.NAME=TN THEN IF (IF G←W3[H←T.[2:10]].[26:4]              00915280 T  0008
                                =TT←W3[F←A.[2:10]].[26:4] THEN NOT CMPA(W2[H],                      00915300 T  0008
                                W2[F]) ELSE G>TT) #,                                                00915320 T  0008
                            NAME = [12:36] # ;                                                      00915340 T  0008
                    J←XRI-1; G←XRI DIV K←XRBUFFDIV3; H←XRBUFF ;                                     00915360 T  0008
                    FOR L←1 STEP 1 UNTIL G DO                                                       00915380 T  0012
                        BEGIN                                                                       00915400 T  0014
                L11:    READ(XREFF,H,XRRY[*]); TRANSFER(W2[T←(L-1)×50],XRRY,K) ;                    00915420 T  0014
                        IJ←T+K-1; TN←-3 ;                                                           00915440 T  0021
                        FOR F←T STEP 1 UNTIL IJ DO                                                  00915460 T  0024
                            BEGIN A[F]←XRRY[TN←TN+3]&F[2:38:10]; W3[F]←XRRY[TN+2] END;               00915480 T  0025
                        END ;                                                                       00915500 T  0030
                    IF H=XRBUFF THEN IF K←XRI MOD K DIV 1≠0 THEN BEGIN H←3×K;GO L11 END;            00915520 T  0035
                    GO L4 ;                                                                         00915540 T  0040
                L1: IF A[K←I].NAME>TN←(T←A[IJ←((L←J)+I+1).[37:10]]).NAME THEN                       00915560 T  0041
                L12:    BEGIN A[IJ]←A[I]; A[I]←T; TN←(T←A[IJ]).NAME END                             00915580 T  0048
                    ELSE IF CMPARGT(A[I]) THEN GO L12 ;                                             00915600 T  0053
                    IF A[J].NAME<TN THEN                                                            00915620 T  0066
                        BEGIN                                                                       00915640 T  0068
                L14:    A[IJ]←A[J]; A[J]←T ;                                                        00915660 T  0068
                        IF TN←(T←A[IJ]).NAME<A[I].NAME THEN                                         00915680 T  0071
                L13:        BEGIN A[IJ]←A[I]; A[I]←T; TN←(T←A[IJ]).NAME END                         00915700 T  0075
                        ELSE IF CMPARGT(A[I]) THEN GO L13 ;                                         00915720 T  0080
                        END                                                                         00915740 T  0093
                    ELSE IF CMPARLS(A[J]) THEN GO L14 ;                                             00915760 T  0093
                L2: IF A[L←L-1].NAME>TN THEN GO L2; IF CMPARGT(A[L]) THEN GO L2 ;                   00915780 T  0106
                L3: IF A[K←K+1].NAME<TN THEN GO L3; IF CMPARLS(A[K]) THEN GO L3 ;                   00915800 T  0122
```

```
    IF K LEQ L THEN BEGIN DOUBLE(A[K],A[L],←,A[L],A[K]); GO L2 END ;
    IF L+K>J+I THEN BEGIN IL[M]←I; IU[M]←L; I←K END
    ELSE BEGIN IL[M]←K; IU[M]←J; J←L END ;
    M←M+1 ;
L4: IF I+10<J THEN GO L1 ;
    IF I=0 THEN IF I<J THEN GO L1 ;
    FOR I←I+1 STEP 1 UNTIL J DO
        IF A[K←I-1].NAME>TN←(T←A[I]).NAME THEN
            BEGIN
L5:         A[K+1]←A[K]; IF A[K←K-1].NAME>TN THEN GO L5 ;
            IF CMPARGT(A[K]) THEN GO L5 ;
            A[K+1]←T ;
            END
        ELSE IF CMPARGT(A[K]) THEN GO L5 ;
    IF (M←M-1) GEQ 0 THEN BEGIN I←IL[M]; J←IU[M]; GO L4 END ;
    G←XRI-1 ;
    FOR I←0 STEP 1 UNTIL G DO
        IF BOOLEAN(L←REAL(A[I]←A[I].NAME&(TN←W3[J←A[I].[2:10]])[1:26:1]
                =XTA)) THEN
            BEGIN
            IF IT←IT+1=9 THEN
                BEGIN LASTLINE←FALSE; WRITE(LINE,15,PRINTBUFF[*]) ;
L6:             BLANKIT(PRINTBUFF,15,IT←0) ;
                END ;
            SETPNT(A[I],W2[J],PRINTBUFF,KLASS[TN.CLASS],TYPES[TN.
                SUBCLASS],IF BOOLEAN(TN) THEN "*" ELSE " ",IT,L-1,
                LASTLINE,6-TN,[27:3]) ;
            END
        ELSE BEGIN
            LASTLINE←TRUE; WRITE(RITE,15,PRINTBUFF[*]); XTA←A[I] ;
            GO L6 ;
            END ;
        WRITE(LINE,15,PRINTBUFF[*]); XRI←0 ;
        END OF XREFCORESORT ;

                PROCEDURE SETUPSTACK ;
PRT(506) = SETUPSTACK
                BEGIN
                   REAL I;

STACK(F+2) = I
                   EMITOPDCLIT(16);
                   EMITL(1);
                   EMITO(ADD);
                   EMITL(16);
                   EMITO(SND);
                   FOR I ← 1 STEP 1 UNTIL LOCALS DO EMITL(0);
                   CHECKINFO;
                   IF DATAPRT ≠ 0 THEN
                        BEGIN ADJUST; EMITOPDCLIT(DATAPRT); EMITO(LNG); EMITB(-1,TRUE);
                              DATASKP←LAX; EMITB(DATASTRT,FALSE); FIXB(DATALINK);
                              EMITL(1); EMITPAIR(DATAPRT,STD); FIXB(DATASKP);
                        END;
                   ADJUST;
                END SETUPSTACK;

                PROCEDURE BRANCHLIT(X,Y); VALUE X,Y; REAL X; BOOLEAN Y;
```

```
00915820 T  0138
00915840 T  0142
00915860 T  0148
00915880 T  0152
00915900 T  0153
00915920 T  0155
00915940 T  0158
00915960 T  0162
00915980 T  0166
00916000 T  0167
00916020 T  0173
00916040 T  0185
00916060 T  0187
00916080 T  0187
00916100 T  0200
00916120 T  0205
00916140 T  0206
00916160 T  0207
00916180 T  0210
00916200 T  0213
00916220 T  0213
00916240 T  0215
00916260 T  0221
00916280 T  0224
00916300 T  0224
00916320 T  0226
00916340 T  0230
00916360 T  0232
00916380 T  0232
00916400 T  0233
00916420 T  0240
00916440 T  0240
00916460 T  0243
00916480 T  0248
34 IS  256 LONG, NEXT SEG    6
00916900 T  0347

00917000 T  0347
00918000 T  0347
START OF SEGMENT **********   35
00919000 T  0000
00920000 T  0000
00921000 T  0001
00922000 T  0002
00923000 T  0003
00924000 T  0003
00925000 T  0008
00925010 T  0008
00925020 T  0009
00925030 T  0012
00925040 T  0015
00925050 T  0018
00925060 T  0018
00926000 T  0018
35 IS   21 LONG, NEXT SEG    6
00927000 T  0347
```

```
PRT(507) = BRANCHLIT
                BEGIN                                                                                  00928000 T  0347
                    IF ADR ≥ 4075 THEN                                                                 00929000 T  0347
                      BEGIN ADR ← ADR+1; SEGOVF END;                                                   00930000 T  0348
                    ADJUST;                                                                            00931000 T  0351
                    IF X.SEGNO≠NSEG THEN BEGIN                                                         00932000 T  0351
                       IF (PRTS+1).[37:2]=1 AND Y THEN                                                 00932300 T  0353
                       EMITOPDCLIT(PRGDESCBLDR(2,0,(ADR+5) DIV 4+1,NSEG))                              00932700 T  0355
                       ELSE EMITOPDCLIT(PRGDESCBLDR(2,0,(ADR+5) DIV 4,NSEG)) END                       00933000 T  0359
                    ELSE                                                                               00934000 T  0364
                    EMITL((ADR+5) DIV 4 - X.LINK DIV 4) ;                                              00935000 T  0364
                END BRANCHLIT;                                                                         00936000 T  0367
                PROCEDURE SEGMENT(SZ,CURSEG,SEGTYP,EDOC); % WRITES OUT EDOC AS SEGMENT                 00937000 T  0368
                    VALUE SZ,CURSEG,SEGTYP;              % UPDATES PDPRT WITH PSUEDO                 00938000 T  0368
                    REAL SZ,CURSEG;                      % UPDATES PDPRT WITH PSUEDO                 00939000 T  0368
                    BOOLEAN SEGTYP;      % TRUE TO WRAP UP A SUBROUTINE BLOCK                        00940000 T  0368
                                         % FALSE TO WRAP UP A SPLIT BLOCK                           00941000 T  0368
                    ARRAY EDOC[0:0];                     % CONTAINS DATA TO WRITE;                   00942000 T  0368
                    BEGIN                                                                              00943000 T  0368
                        STREAM PROCEDURE M1(F,T); BEGIN DI ← T; SI ← F; DS ← 2 WDS END;                00944000 T  0368
                                                                                  START OF SEGMENT **********   36
PRT(510) = M1
                        REAL T;                                                                        00945000 T  0000
STACK(F+2) = T
                        REAL BEGINSUB, ENDSUB, HERE;                                                   00946000 T  0001
STACK(F+3) = BEGINSUB
STACK(F+4) = ENDSUB
STACK(F+5) = HERE
                        LABEL WRITEPGM;                                                                00947000 T  0001
                        INTEGER I, CNTR;                                                               00948000 T  0001
STACK(F+6) = I
STACK(F+7) = CNTR
                    IF DEBUGTOG THEN FLAGROUTINE(" SEGM","ENT   ",TRUE ) ;                             00948010 T  0001
                        IF NOT SEGTYP  THEN GO TO WRITEPGM;                                            00949000 T  0003
                        IF SPLINK > 1 AND NOT RETURNFOUND THEN                                         00949100 T  0003
                        BEGIN XTA ← BLANKS; FLAG(142) END;                                             00949200 T  0005
                        IF SPLINK < 0 THEN                                                             00950000 T  0007
                    BEGIN                                                                              00951000 T  0008
                        ADJUST;                                                                        00952000 T  0009
                        BDPRT[BDX,BDX+1] ← PRGDESCBLDR(1, 0, (ADR+1) DIV 4, NSEG);                     00953000 T  0009
                        FOR I ← 1 STEP 1 UNTIL LOCALS DO EMITL(0);                                     00954000 T  0014
                        CHECKINFO;                                                                     00955000 T  0020
                                                                                                       00955500 T  0020
                        EMITB(0&INITIALSEGNO[TOSEGNO], FALSE);                                         00956000 T  0020
                    END                                                                                00957000 T  0022
                        ELSE                                                                           00958000 T  0022
                        IF SPLINK = 1 THEN              % MAIN PROGRAM                                  00959000 T  0022
                    BEGIN                                                                              00960000 T  0023
                        ADJUST;                                                                        00961000 T  0024
                        IF STRTSEG ≠ 0 THEN FLAG(75);                                                  00962000 T  0024
                        STRTSEG ← NSEG &                                                               00963000 T  0026
                           PRGDESCBLDR(1, 0, (ADR+1) DIV 4, NSEG)[18:33:15];                           00964000 T  0027
                        SETUPSTACK;                                                                    00965000 T  0030
                                                                                                       00965500 T  0030
                        EMITB(0&INITIALSEGNO[TOSEGNO], FALSE);                                         00966000 T  0030
                    END                                                                                00967000 T  0032
                        ELSE                                                                           00968000 T  0032
```

```
        IF ELX ≤ 1 THEN                                                                                  00969000 T  0032
    BEGIN                                                                                                00970000 T  0034
        ADJUST;                                                                                          00971000 T  0034
        T ← PRGDESCBLDR(1, GET(SPLINK).ADDR, (ADR+1) DIV 4, NSEG);                                       00972000 T  0035
        SETUPSTACK;                                                                                      00973000 T  0039
        FIXPARAMS(0);                                                                                    00974000 T  0039
        INFO[SPLINK.IR, SPLINK.IC].SEGNO ← NSEG;                                                         00975000 T  0040
    END                                                                                                  00976000 T  0044
        ELSE                                                                                             00977000 T  0044
    BEGIN                                                                                                00978000 T  0044
        ADJUST;                                                                                          00979000 T  0045
        BEGINSUB ← (ADR+1) & NSEG[TOSEGNO];                                                              00980000 T  0045
        EMITL(17); EMITO(STD);                                                                           00981000 T  0047
        SETUPSTACK;                                                                                      00982000 T  0049
                                                                                                         00982100 T  0049
        EMITOPDCLIT(17); EMITO(GFW);                                                                     00983000 T  0049
        ENDSUB ← ADR & NSEG[TOSEGNO];                                                                    00984000 T  0051
        FOR I ← 0 STEP 1 UNTIL ELX-1 DO                                                                  00985000 T  0053
    BEGIN                                                                                                00986000 T  0057
        ADJUST;                                                                                          00987000 T  0057
        HERE ← (ADR+1) DIV 4;                                                                            00988000 T  0057
        T ← ENTRYLINK[I].LINK;                                                                           00989000 T  0059
%VOID                                                                                                    00990000 T  0061
        T ← PRGDESCBLDR(1, GET(T).ADDR, HERE, NSEG);                                                     00991000 T  0061
        BRANCHLIT(ENDSUB,FALSE);                                                                         00992000 T  0064
        EMITB(BEGINSUB, FALSE);                                                                          00993000 T  0065
      ADJUST;                                                                                            00994000 T  0066
        FIXPARAMS(I);                                                                                    00995000 T  0066
        INFO[(ENTRYLINK[I].LINK).IR,(ENTRYLINK[I].LINK).IC].SEGNO←NSEG;                                  00995500 T  0067
    END;                                                                                                 00996000 T  0071
    END;                                                                                                 00997000 T  0073
                                                                                                         00998000 T  0073
        SZ ← (ADR+4) DIV 4;                                                                              00999000 T  0075
        CNTR ← 0;                                                                                        00999100 T  0076
        CURSEG ← NSEG;                                                                                   01000000 T  0076
        WRITEPGM:                                                                                        01001000 T  0077
        NSEG ← ((T ← SZ) + 29) DIV 30;                                                                   01002000 T  0079
        IF DALOC DIV CHUNK < I ← (DALOC+NSEG) DIV CHUNK                                                  01003000 T  0081
         THEN DALOC ← CHUNK × I;    % INSURE SEGMENT DONT BREAK                                           01004000 T  0083
                                    % ACROSS ROW                                                         01005000 T  0083
        IF LISTOG THEN WRITALIST(SFGEND,2,CURSEG,T,0,0,0,0,0,0) ;                                        01006000 T  0088
        PDPRT[PDIR,PDIC] ←                  % PDPRT ENTRY FOR SEGMENT                                     01007000 T  0090
           SZ&DALOC[DKAC]                                                                                01007100 T  0092
             & GET(SPLINK)[12:14:1]                                                                      01008000 T  0093
             &CURSEG[SGNOC];                                                                             01009000 T  0095
        IF ERRORCT = 0 THEN                                                                              01010000 T  0095
        DO BEGIN                                                                                         01011000 T  0097
        FOR I←0 STEP 2 WHILE I < 30 AND CNTR < SZ DO                                                     01012000 T  0102
        BEGIN M1(FDOC[CNTR.[38:3],CNTR.[41:7]],CODE(I));                                                 01013000 T  0107
              CNTR ← CNTR + 2;                                                                           01014000 T  0109
        END;                                                                                             01015000 T  0109
        WRITE(CODE[DALOC]);                                                                              01016000 T  0114
        DALOC ← DALOC +1;                                                                                01017000 T  0115
        END UNTIL CNTR ≥ SZ;                                                                             01018000 T  0116
        PDINX ← PDINX +1;                                                                                01018025 T  0118
        IF NOT SEGOVFLAG THEN                                                                            01018100 T  0119
        IF PXREF THEN                                                                                    01018110 T  0120
        IF(EODS←(NEXT=EOF))AND NOT XGLOBALS THEN ELSE
```

```
                        BEGIN KLASS[6] ← "LABEL ";                                          01018120 T  0123
                              WRITE(XREFF,XRBUFF,XRRY[*]) ;                                 01018125 T  0125
                              IF FIRSTCALL THEN DATIME ELSE WRITE(PTR[PAGE]);               01018130 T  0129
                               IF SPLINK<0 THEN                                             01018135 T  0137
                                  BEGIN WRITE(PTR,XHEDB);REWIND(XREFF);END                  01018140 T  0137
                               ELSE                                                         01018145 T  0143
                              IF FODS THEN BEGIN WRITE(PTR,XHEDG); REWIND(XREFG) END        01018150 T  0143
                              ELSE BEGIN REWIND(XREFF); C2←(XR[4].SUBCLASS×2+5);            01018160 T  0149
                                         IF IT←XR[4].CLASS=FUNID THEN WRITE(PTR,XHEDF,      01018170 T  0153
PRT(511) = *LIST, LABEL, OR SEGMENT DESCRIPTOR*
                                           XR[C2],XR[C2+1],XR[3],XR[C2+12],                 01018180 T  0158
                                           XR[C2+13],"------")                              01018190 T  0167
                                         ELSE IF IT=SUBRID THEN WRITE(PTR,XHEDS,XR[3],      01018200 T  0170
PRT(512) = *LIST, LABEL, OR SEGMENT DESCRIPTOR*
                                           "------") ELSE WRITE(PTR,XHEDM);                 01018210 T  0183
                                END;                                                        01018220 T  0192
                              IT←XTA←0 ;                                                    01018260 T  0192
                              LASTLINE ← FALSE;                                             01018280 T  0194
                              BLANKIT(PRINTBUFF,15,0);                                      01018320 T  0195
                              IF XRI>1023 OR EODS THEN SORT(PT,INP,0,HV,CMP,3,4000)         01018330 T  0197
PRT(513) =      0101833000000000304206600017ERRORCT000000304
PRT(514) = MERGE
                              ELSE XREFCORESORT ;                                           01018340 T  0218
PRT(515) = SORT
                              REWIND(XREFF);                                                01018350 T  0221
                              PXREF←IF XREF THEN TRUE ELSE FALSE;                           01018355 T  0223
                              KLASS[6] ← "ERROR ";                                          01018360 T  0226
                              IF (NOT SEGTYP  AND EODS) OR (SEGTYP  AND LISTOG AND          01018370 T  0227
                                  NOT SEGPTOG) THEN WRITE (PTR[PAGE]);                      01018375 T  0230
                        END;                                                                01018380 T  0236
                        IF LISTOG AND SEGTYP AND SEGPTOG THEN WRITE(PTR[PAGE]);             01019000 T  0236
                        IF SEGTYP THEN                                                      01019100 T  0242
                           BEGIN                                                            01019110 T  0243
                           FOR I←12,17 STEP 1 UNTIL 24,39 STEP 1 UNTIL 41,                  01019150 T  0243
                           50 STEP 1 UNTIL 57 DO TIPE[I]←REALID ;                           01019160 T  0253
STACK(F+10) = *TEMPORARY STORAGE*
                           FOR I←25,33 STEP 1 UNTIL 37 DO TIPE[I]←INTID ;                   01019170 T  0260
STACK(F+11) = *TEMPORARY STORAGE*
                           LASTNEXT←1000 ;                                                  01019190 T  0267
                           END ;                                                            01019200 T  0268
                        ENDSEGTOG ← TRUE;                                                   01020000 T  0268
                        TSEGSZ ← TSEGSZ + SZ;                                               01021000 T  0270
                        COMMENT IF SEGSW THEN LETS ALSO WRITE OUT THE LINE SEGMENTS         01021010 T  0271
                              THAT WE VE GONE TO SUCH TROUBLE BUILDING. LINESEG             01021150 T  0271
                              CONTAINS A CARD SEQUENCE NUMBER(BINARY) IN [10:28]            01021200 T  0271
                              AND THE WORD BOUNDARY ADDRESS OF THE FIRST CODE               01021250 T  0271
                              SYLLABLE IN [38:10];                                          01021300 T  0271
                        IF SEGSW THEN                                                       01021350 T  0271
                        IF NOLIN > 0 THEN                                                   01021360 T  0272
                        BEGIN                                                               01021400 T  0273
                           LINEDICT[CURSEG.IR,CURSEG.IC] ← % UPDATE LINE DICTIONARY         01021450 T  0273
                            0 & NOLIN[18:33:15] & DALOC[33:33:15];%FOR THIS SEGMENT         01021500 T  0276
                           CNTR ← 0;  DO BEGIN                                              01021550 T  0277
                           FOR I ← 0 STEP 2 WHILE I<30 AND CNTR<NOLIN DO                    01021600 T  0279
                           BEGIN                                                            01021650 T  0284
                              M1(LINESEG[CNTR,[38:3],CNTR,[41:7]],CODE(I));                 01021700 T  0284
                              CNTR ← CNTR + 2                                               01021750 T  0289
```

```
                        END;
                        WRITE(CODE[DALOC]);
                        DALOC ← DALOC + 1;
                        END UNTIL CNTR ≥ NOLIN;
                    END ;
                    IF NOT SEGOVFLAG THEN XRI != 0;
                    IF DEBUGTOG THEN FLAGROUTINE("  SEGM","ENT    ",FALSE) ;
                END SEGMENT;

            PROCEDURE WRITEDATA(SZ,SEG,ARY);  % WRITES OUT TYPE 2 SEGMENTS
                VALUE SZ,SEG;
                REAL SZ,SEG;
                ARRAY ARY[0];
                BEGIN
                    INTEGER T,NSEG,I,CNTR;
STACK(F+2) = T
STACK(F+3) = NSEG
STACK(F+4) = I
STACK(F+5) = CNTR
                    INTEGER TYPE;
STACK(F+6) = TYPE
                    STREAM PROCEDURE M30(F,T,SZ); VALUE SZ;
PRT(516) = M30
                    BEGIN
                        SI ← F; DI ← T; DS ← SZ WDS;
                    END M30;
                    IF SZ ≥ 0 THEN TYPE ← 2 ELSE SZ ← -SZ;
                    NSEG ← (( T ← SZ) + 29) DIV 30;
                    IF DALOC DIV CHUNK < I ← (DALOC+NSEG) DIV CHUNK
                      THEN DALOC ← CHUNK × I;
                    PDPRT[PDIR,PDIC] ←
                      SZ&DALOC[DKAC]
                        &TYPE[STYPC]
                        &SEG[SGNOC];
                    PDINX ← PDINX + 1;
                    IF ERRORCT = 0 THEN
                    FOR I ← 0 STEP 30 WHILE I < SZ DO
                BEGIN
                    M30(ARY[I],CODE(0),IF (SZ-I) > 30 THEN 30 ELSE (SZ-I));
                    WRITE(CODE[DALOC]);
                    DALOC ← DALOC + 1;
                END;
                    IF LISTOG THEN WRITALIST(SEGEND,2,SEG,T,0,0,0,0,0,0) ;
                    TSEGSZ ← TSEGSZ + SZ;
                END WRITEDATA;

            REAL PROCEDURE PRGDESCBLDR(DT,PRT,RELADR,SGNO);
                VALUE DT,PRT,RELADR,SGNO;
                REAL  DT,PRT,RELADR,SGNO;
                BEGIN
                FORMAT FMT("PRT=",A4,", REL-ADR=",A4,", SEG=",I4,", TYPE=",A2);
PRT(517) = FMT
                IF PRT=0 THEN BEGIN BUMPPRT; PRT←PRTS END;
```

```
01021800 T  0290
01021850 T  0291
01021900 T  0296
01021950 T  0297
01021955 T  0298
01021957 T  0298
01021959 T  0301
01022000 T  0303
36 IS  309 LONG, NEXT SEG    6
01023000 T  0368
01024000 T  0368
01025000 T  0368
01026000 T  0368
01027000 T  0368
01028000 T  0368
START OF SEGMENT **********   37

01029000 T  0000
01030000 T  0000
01031000 T  0000
01032000 T  0000
01033000 T  0001
01034000 T  0001
01035000 T  0005
01036000 T  0007
01037000 T  0009
01038000 T  0012
01039000 T  0014
01040000 T  0015
01041000 T  0016
01042000 T  0018
01043000 T  0019
01044000 T  0020
01045000 T  0024
01046000 T  0024
01047000 T  0032
01048000 T  0036
01049000 T  0038
01050000 T  0038
01051000 T  0043
01052000 T  0044
37 IS   48 LONG, NEXT SEG    6
01053000 T  0368
01054000 T  0368
01055000 T  0368
01056000 T  0368
01057000 T  0368
START OF SEGMENT **********   38
START OF SEGMENT **********   39

39 IS   13 LONG, NEXT SEG   38
01058000 T  0000
```

```
            PDPRT[PDIR,PDIC] ←                                                    01059000 T  0005
               O&DT[DTYPC]                                                        01060000 T  0008
                &PRT[PRTAC]                                                       01061000 T  0009
                &RELADR[RELADC]                                                   01062000 T  0010
                &SGNO[SGNOC];                                                     01063000 T  0011
            PDINX ← PDINX + 1;                                                    01064000 T  0012
            IF CODETOG THEN WRITALIST(FMT,4,B2D(PRT),B2D(RELADR),SGNO,            01065000 T  0014
            IF DT=0 THEN "AE" ELSE IF DT=1 THEN "PD" ELSE "LD",0,0,0,0) ;         01066000 T  0017
            PRGDESCBLDR ← PRT;                                                    01067000 T  0023
        END PRGDESCBLDR;                                                          01068000 T  0024
                                                                  38 IS   30 LONG, NEXT SEG    6
    PROCEDURE EQUIV(R); VALUE R; REAL R;                                          01069000 T  0368
    COMMENT  THIS PROCEDURE FIXES UP THE INFO TABLE FOR THE EQUIV OR              01069100 T  0368
               COMMON RING. THE FIRST ELEMENT PAST HAS AN OFFSET (DO NOT          01069110 T  0368
               ALTER THIS) THE TYPE IS FIXED.  THE OFFSET IS DETERMINED           01069120 T  0368
               FROM THE FIRST.  CORRECTINFO ADJUST THE OFFSET IF THERE            01069130 T  0368
               IS A NEGATIVE OFFSET ON ANY ELEMENT.  THE INFA[ADJ] BIT IS         01069140 T  0368
               SET IF THE ELEMENT HAS A NEGATIVE OFFSET.  IF THE ELEMENT          01069150 T  0368
               APPEARED IN MORE THAN ONE EQUIVALENCE STATEMENT OR AN              01069160 T  0368
               EQUIVALENCE STATEMENT AND A COMMON STATEMENT THE ELEMENTS          01069170 T  0368
               ARE LINKED BY COM[LASTC] WHICH POINTS TO THE HEADER                01069180 T  0368
               OF THAT STATEMENT;                                                 01069190 T  0368
    BEGIN                                                                         01070000 T  0368
       DEFINE BASS     = LOCALS     #,  % THESE DEFINES ARE USED TO REDUCE THE    01071000 T  0368
                                                          START OF SEGMENT **********   40
              REL      = PARMS      #,  % STACKSIZE OF EQUIV FOR RECURSION        01071100 T  0000
              I        = PRTS       #,                                            01071200 T  0000
              T        = LSTS       #,                                            01071250 T  0000
              Q        = LSTA       #,                                            01071400 T  0000
              LAST     = TV         #,                                            01071500 T  0000
              B        = SAVESUBS   #,                                            01071600 T  0000
              P        = NAMEIND    #,                                            01071700 T  0000
              PRTX     = LSTI       #,                                            01071800 T  0000
              C        = FX1        #,                                            01071900 T  0000
              INFA     = FX2        #,                                            01072000 T  0000
              INFB     = FX3        #,                                            01072100 T  0000
              INFC     = NX1        #;                                            01072200 T  0000
       LABEL XIT, CHECK ;                                                         01073000 T  0000
    IF DEBUGTOG THEN FLAGROUTINE("   EQU","IV    ",TRUE ) ;                       01073010 T  0000
       IF GETC(R)<0 THEN GO TO XIT ;                                              01074000 T  0002
       PUTC(R,-GETC(R)) ;                                                         01075000 T  0003
       PRTX ← GROUPPRT;                                                           01076000 T  0005
       C←REAL(GETC(R),CE=1) ;                                                     01077000 T  0006
       LAST←GETC(R),LASTC=1 ;                                                     01078000 T  0008
       BASS←GETC(R+1); P←0 ;                                                      01079000 T  0010
       FOR I ← R+2 STEP 1 UNTIL LAST DO                                           01080000 T  0013
       BEGIN IF GETC(I).CLASS=ENDCOM THEN I←GETC(I).LINK ;                        01081000 T  0017
       GETALL(T←GETC(I).LINK,INFA,INFB,INFC) ;                                    01082000 T  0021
       IF Q←INFC.SIZE=0 THEN %   A SIMPLE VARIABLE                                01083000 T  0024
          INFC.SIZE ← Q ← IF INFA.SUBCLASS ≤ LOGTYPE THEN 1 ELSE 2;               01084000 T  0026
       PUT(T+2,INFC);                                                             01085000 T  0031
       IF INFA.SUBCLASS>LOGTYPE THEN        SEENADOUB←TRUE;                       01085500 T  0032
       IF BOOLEAN(C) THEN                                                         01086000 T  0036
          BEGIN COM[PWI].RELADD←REL←P ;                                           01087000 T  0036
             P ← P + Q;                                                           01088000 T  0041
          END ELSE COM[PWI].RELADD←REL←BASS+GETC(I).RELADD ;                       01089000 T  0042
          IF INFA    < 0 THEN IF INFA    .ADJ = 1 THEN                            01090000 T  0052
```

```
      B ← -INFC.BASE - REL ELSE B ← INFC.BASE - REL;                                01091000 T  0054
END;                                                                               01092000 T  0059
FOR I ← R+2 STEP 1 UNTIL LAST DO                                                   01093000 T  0060
BEGIN IF GETC(I).CLASS=ENDCOM THEN I←GETC(I).LINK ;                                01094000 T  0064
   GETALL(T←GETC(I).LINK,INFA,INFB,INFC) ;                                         01095000 T  0068
IF INFA.CLASS = 1 THEN SWARYCT ← 1;                                                01095500 T  0071
   P←B+GETC(I).RELADD ;                                                            01096000 T  0073
   Q ← INFC    .SIZE;                                                              01097000 T  0076
   IF INFA     < 0 THEN                                                            01098000 T  0077
   BEGIN IF INFA    .ADJ = 1 THEN BASS ← -INFC    .BASE ELSE                       01099000 T  0078
      BASS ← INFC.BASE;                                                            01100000 T  0081
      IF P ≠ BASS THEN                                                             01101000 T  0083
      BEGIN XTA ← INFB    ; FLAG(2) END;                                           01102000 T  0084
      GO TO CHECK;                                                                 01104000 T  0086
   END;                                                                            01105000 T  0086
   INFA     ← -INFA     & PRTXTTOADDR];                                            01108000 T  0086
IF INFA    .CLASS = UNKNOWN THEN INFA    .CLASS ← VARID;                          01109000 T  0088
INFA    .TYPEFIXED ← 1;                                                            01110000 T  0092
INFC.BASE ← P;                                                                     01111000 T  0094
PUT(T+2,INFC);                                                                     01112000 T  0095
IF P < 0 THEN INFA    .ADJ ← 1;                                                    01113000 T  0096
PUT(T,INFA);                                                                       01114000 T  0099
CHECK;                                                                             01115000 T  0100
IF P+Q > LENGTH THEN LENGTH ← P+Q;                                                 01116000 T  0101
IF P < LOWERBOUND THEN LOWERBOUND ← P;                                             01117000 T  0104
END;                                                                               01118000 T  0106
FOR I ← R+2 STEP 1 UNTIL LAST DO                                                   01119000 T  0106
BEGIN                                                                              01120000 T  0110
   IF GETC(I).CLASS=ENDCOM THEN I←GETC(I).LINK ;                                   01121000 T  0110
   IF T←GETC(I).LASTC≠R THEN                                                       01122000 T  0114
        IF GETC(T)≥0 THEN                                                          01122500 T  0117
            BEGIN R←R&LAST[3:33:15]&I[18:33:15] ;                                  01122550 T  0118
            EQUIV(T);  LAST←R.[3:15];  I←R.[18:15];  R←R.[33:15] ;                 01122600 T  0121
            END ;                                                                  01122650 T  0126
      END;                                                                         01123000 T  0126
      XIT;                                                                         01124000 T  0126
   IF DEBUGTOG THEN FLAGROUTINE("   EQU","IV    ",FALSE) ;                       01124010 T  0127
   END EQUIV;                                                                      01125000 T  0129
                                                                          40 IS  132 LONG, NEXT SEG     6
   ALPHA PROCEDURE GETSPACE(S); VALUE S; ALPHA S;                                  01126000 T  0368
   BEGIN LABEL RPLUS, FMINUS, XIT;                                                 01127000 T  0368
                                                                       START OF SEGMENT **********    41
      LABEL DONE;                                                                  01128000 T  0000
      REAL A;                                                                      01129000 T  0000
STACK(F+3) = A
      BOOLEAN OWNID;                                                               01129100 T  0000
STACK(F+4) = OWNID
      IF OWNID ← S < 0 THEN S ← -S;        % IN DATA STMT, THUS OWN               01129200 T  0000
      IF A ← GET(GETSPACE←S) < 0 THEN GO TO DONE;                                  01130000 T  0002
      IF A.CLASS GEQ 13 THEN FLAG(34) ELSE                          %104-          01130500 C  0005
   CASE A.CLASS OF                                                                 01131000 T  0008
   BEGIN                                                                           01132000 T  0009
PRT(520) = *CASE STATEMENT DESCRIPTOR*
   BEGIN                                                                           01133000 T  0009
      PUT(S,A←A&VARID[TOCLASS]);                                                   01134000 T  0009
      PUT(S+2,(GET(S+2)   &(IF A.SUBCLASS ≤ LOGTYPE                                01135000 T  0012
          THEN 1 ELSE 2 )[TOSIZE]));                                               01136000 T  0015
```

```
FND;
   IF BOOLEAN(A,FORMAL) THEN BUMPLOCALS;
;
BEGIN A.TYPEFIXED ← 1; GO TO RPLUS END;
GO TO RPLUS;
GO TO RPLUS;
GO TO DONE;
BEGIN A.TYPEFIXED ← 1; IF BOOLEAN(A.FORMAL) THEN GO TO FMINUS
   ELSE GO TO RPLUS;
END;
BEGIN A.TYPEFIXED ← 1; GO TO RPLUS END;
IF BOOLEAN(A,FORMAL) THEN GO TO FMINUS ELSE GO TO RPLUS;
IF BOOLEAN(A,FORMAL) THEN GO TO FMINUS ELSE GO TO RPLUS;
GO TO RPLUS;
GO TO RPLUS;
END OF CASE STATEMENT;


A,TYPEFIXED ← 1;
IF BOOLEAN(A,FORMAL) THEN
        GO TO FMINUS;
IF BOOLEAN(A,CE) OR BOOLEAN(A,EQ) THEN
BEGIN
   PUT(S,A);
   CALLEQUIV(A,ADDR,OWNID,GET(S+1)) ;
   GO TO DONE;
END;
   A,TWOD ← REAL(GET(S+2),SIZE > 1023);
FMINUS:
   IF OWNID THEN BEGIN BUMPPRT; A,ADDR←PRTS END ELSE
    BEGIN BUMPLOCALS; A,ADDR ← LOCALS +1536 END;
   IF A.CLASS = VARID THEN IF A.SUBCLASS ≥ DOUBTYPE THEN
   IF OWNID THEN BUMPPRT ELSE
   BUMPLOCALS;
   GO TO XIT;
RPLUS:
   BUMPPRT; A,ADDR←PRTS;
XIT:
   PUT(S, -A);
DONE:
END GETSPACE;

                INTEGER STREAM PROCEDURE LBLSHFT(S);     VALUE S;
PRT(521) = LBLSHFT
                BEGIN
                     LOCAL T;
                     LABEL L;
                     DI ← LOC LBLSHFT; DS ← 8 LIT "00      ";
                     DI ← DI - 6; SI ←LOC S; SI ← SI + 2;
                     TALLY ← 1; T ← TALLY;
                 SCT(IF SC="0" THEN BEGIN SI←SI+1; JUMP OUT 1 TO L END
                  ELSE TALLY←0; T←TALLY);
                     IF SC≥"0" THEN DS←CHR ELSE IF SC=" " THEN SI←SI+1
                     ELSE JUMP OUT;  L: ) ;
                     IF SC ≠ " " THEN BEGIN DI ← LOC LBLSHFT; DS ← LIT "+" END;
                END LBLSHFT;
                     COMMENT EMITTERS AND CODE CONTROL;
```

```
01137000 T  0017
01138000 T  0018
01139000 T  0023
01140000 T  0023
01141000 T  0026
01142000 T  0027
01143000 T  0027
01144000 T  0028
01145000 T  0030
01146000 T  0031
01147000 T  0032
01148000 T  0034
01149000 T  0037
01150000 T  0039
01151000 T  0039
01152000 T  0040
START OF SEGMENT **********    42
  42 IS    14 LONG, NEXT SEG    41
01153000 T  0040
01154000 T  0042
01155000 T  0043
01156000 T  0044
01157000 T  0045
01158000 T  0046
01159000 T  0047
01160000 T  0050
01161000 T  0050
01162000 T  0050
01163000 T  0054
01163100 T  0055
01164000 T  0061
01165000 T  0067
01165100 T  0070
01166000 T  0075
01167000 T  0081
01168000 T  0082
01169000 T  0083
01170000 T  0088
01171000 T  0089
01172000 T  0090
01173000 T  0091
41 IS    94 LONG, NEXT SEG     6
01174000 T  0368
01175000 T  0368
01176000 T  0368
01177000 T  0368
01178000 T  0368
01179000 T  0369
01180000 T  0370
01181000 T  0370
01182000 T  0373
01183000 T  0374
01183100 T  0376
01184000 T  0379
01185000 T  0380
01186000 T  0381
```

```
                ALPHA PROCEDURE B2D(B); VALUE B; REAL B;                                    01187000 T  0381
                      B2D ← 0&B[45:45:3]&B[39:42:3]&B[33:39:3]&B[27:36:3];                   01188000 T  0381
                PROCEDURE DEBUG(S);     % PRINTS OUT DEBUG CODE                            01189000 T  0389
PRT(522) = DEBUG
                    VALUE S; REAL S;    % IF S<0 THEN S IS FIELD TYPE OPERATOR              01190000 T  0389
                    BEGIN                                                                   01191000 T  0389
                        FORMAT FF(X35,*(33(".")),A4,":",A1,2(X2,A4),X4,A4) ;                01192000 T  0389
                                                                                  START OF SEGMENT **********   43
                                                                                  START OF SEGMENT **********   44
PRT(523) = FF
                                                                                    44 IS   18 LONG, NEXT SEG   43
                        ALPHA CODE,MNM,SYL;                                             01193000 T  0000
STACK(F+2) = CODE
STACK(F+3) = MNM
STACK(F+4) = SYL
                    REAL T;                                                             01194000 T  0000
STACK(F+5) = T
                PROCEDURE SEARCH(CODE,S); VALUE S; REAL S,CODE;                         01195000 T  0000
PRT(524) = SEARCH
                    BEGIN  % SEARCHS WOP TO FIND CODE FOR S                             01196000 T  0000
                    REAL N,I;                                                           01197000 T  0000
                                                                                  START OF SEGMENT **********   45
STACK(F+2) = N
STACK(F+3) = I
                    LABEL L;                                                            01198000 T  0000
                        N ← 64;                                                         01199000 T  0000
                        FOR I ← 66 STEP IF WOP[I] < S THEN N ELSE -N                    01200000 T  0000
                        WHILE N ← N DIV 2 ≥ 1 DO                                        01201000 T  0005
                        IF WOP[I] =S THEN GO TO L;                                      01202000 T  0009
                        I ← 0;          % NOT FOUND                                    01203000 T  0011
                    L:  CODE ← WOP[I+1];                                                01204000 T  0011
                END SEARCH;                                                             01205000 T  0013
                                                                                    45 IS   16 LONG, NEXT SEG   43
                        IF S < 0 THEN                                                   01206000 T  0000
                    BEGIN       % FIELD TYPE OPERATOR                                   01207000 T  0000
                        SYL ← S;                                                        01208000 T  0001
                        MNM ← B2D(S,[36:6]);                                            01209000 T  0002
                        IF (S ← S.[42:6]) = 37 THEN CODE ← "ISO " ELSE                  01210000 T  0003
                        IF S = 45 THEN CODE ← "DIA " ELSE                               01211000 T  0006
                        IF S = 49 THEN CODE ← "DIB " ELSE                               01212000 T  0011
                        IF S = 53 THEN CODE ← "TRB ";                                   01213000 T  0015
                    END                                                                 01214000 T  0019
                        ELSE                                                            01215000 T  0019
                    BEGIN                                                               01216000 T  0019
                        IF (T ← S.[46:2]) ≠ 1 THEN                                      01217000 T  0021
                    BEGIN                                                               01218000 T  0022
                        SYL ← S;                                                        01219000 T  0023
                        MNM ← B2D(S,[36:10]);                                           01220000 T  0024
                        IF T = 0 THEN CODE ← "LITC"                                     01221000 T  0025
                        ELSE IF T =2 THEN CODE ← "OPDC"                                 01222000 T  0027
                        ELSE CODE ← "DESC";                                             01223000 T  0031
                    END                                                                 01224000 T  0034
                        ELSE                                                            01225000 T  0034
                    BEGIN       % SEARCH WOP FOR OPERATOR NAME                          01226000 T  0034
                        SYL ←S;                                                         01227000 T  0037
                        MNM ← "    ";                                                   01228000 T  0037
                        SEARCH(CODE,S.[36:10]);                                         01229000 T  0038
```

```
                    END;
                    END;
                         WRITALIST(FF,6,IF ADR≤DEBUGADR THEN 1 ELSE -1,B2D(ADR,[36:10]),
                              B2D(ADR,[46:2]),CODE,MNM,B2D(SYL),0,0) ;
                         IF DEBUGADR<ADR THEN DEBUGADR←ADR ;
                    END DEBUG;

               PROCEDURE DEBUGWORD(N); VALUE N; REAL N; %PRINTS OUT C+ CONSTANTS
                    BEGIN
                         STREAM PROCEDURE WB2D(F,T);
PRT(525) = WB2D
                              BEGIN
                                   DI ← T;
                                   DI←DI+35; DS←10LIT"CONSTANT  " ;
                                   SI ← F;
                                   16(DS ← 3 RESET; 3(IF SB THEN DS ← SET
                                                      ELSE DS ← RESET;
                                                 SKIP SB));
                              END WB2D;
                         ARRAY A[0:14];  FORMAT F(X63,"(DEC ",B,")") ;
STACK(F+2) = A

PRT(526) = F

                         WRITE(A[*],F,N);  WB2D(N,A) ;
PRT(527) = *LIST, LABEL, OR SEGMENT DESCRIPTOR*
                         WRITAROW(15,A) ;
                    END DEBUGWORD;

               REAL PROCEDURE GIT(Z);  % RETURNS OPERATOR IN SYLLABLE Z
PRT(530) = GIT
                    VALUE Z; REAL Z;
                    BEGIN
                         INTEGER STREAM PROCEDURE GT(S,SKP); VALUE SKP;
PRT(531) = GT
                         BEGIN
                              SI ← S; SKP(SI ← SI + 2);
                              DI ← LOC GT; DI ← DI + 6; DS ← 2 CHR;
                         END GT;
                         GIT ← GT(FDOC[Z,[36:3],Z,[39:7]],Z,[46:2]);
                    END GIT;

               REAL STREAM PROCEDURE SPCLBIN2DEC(N);  VALUE N ;
PRT(532) = SPCLBIN2DEC
                    BEGIN LOCAL L; LABEL B1 ;
                    DI←LOC L; SI←LOC N; DS←8DEC; SI←LOC L; TALLY←8 ;
               B1:  IF SC="0" THEN BEGIN TALLY←TALLY+63; SI←SI+1; GO B1 END ;
                    DI←LOC SPCLBIN2DEC; DI←DI+2; DS←6LIT" "; DI←DI-6; N←TALLY; DS←N CHR;
                    END OF SPCLBIN2DEC ;
               STREAM PROCEDURE PACK(T,F,SKP); VALUE SKP,F;
PRT(533) = PACK
                    BEGIN     % PACKS OPERATORS INTO EDOC
                         SI ← LOC F; SI ← SI + 6;DI ← T; SKP(DI ← DI + 2);
                         DS ← 2 CHR;
                    END PACK;
```

```
                                                                  01230000 T  0040
                                                                  01231000 T  0040
                                                                  01232000 T  0040
                                                                  01233000 T  0044
                                                                  01233100 T  0048
                                                                  01234000 T  0050
                                                          43 IS    54 LONG, NEXT SEG    6
                                                                  01235000 T  0389
                                                                  01236000 T  0389
                                                                  01237000 T  0389
                                                         START OF SEGMENT **********   46

                                                                  01238000 T  0000
                                                                  01239000 T  0000
                                                                  01240000 T  0000
                                                                  01241000 T  0002
                                                                  01242000 T  0002
                                                                  01243000 T  0003
                                                                  01244000 T  0004
                                                                  01247000 T  0005
                                                                  01248000 T  0005

                                                         START OF SEGMENT **********   47

                                                          47 IS     7 LONG, NEXT SEG   46
                                                                  01249000 T  0007

                                                                  01250000 T  0017
                                                                  01251000 T  0019

                                                          46 IS    23 LONG, NEXT SEG    6
                                                                  01252000 T  0389

                                                                  01253000 T  0389
                                                                  01254000 T  0389
                                                                  01255000 T  0389
                                                         START OF SEGMENT **********   48

                                                                  01256000 T  0000
                                                                  01257000 T  0000
                                                                  01258000 T  0001
                                                                  01259000 T  0002
                                                                  01260000 T  0003
                                                                  01261000 T  0008
                                                          48 IS    11 LONG, NEXT SEG    6
                                                                  01261100 T  0389

                                                                  01261110 T  0389
                                                                  01261120 T  0390
                                                                  01261130 T  0391
                                                                  01261140 T  0393
                                                                  01261150 T  0396
                                                                  01262000 T  0397

                                                                  01263000 T  0397
                                                                  01264000 T  0398
                                                                  01265000 T  0400
                                                                  01266000 T  0400
```

```
                    PROCEDURE EMITO(N); VALUE N; REAL N;                                    01267000 T  0400
                        BEGIN                                                               01268000 T  0400
                       BUMPADR;                                                             01269000 T  0400
                             PACK(EDOC[EDOCI],(N←1&N[36:38:10]),ADR,[46:2]);                01270000 T  0403
                             IF CODETOG THEN DEBUG(N);                                      01271000 T  0409
                    END EMITO;                                                              01272000 T  0410
                    PROCEDURE EMITN(P); VALUE P; REAL P;                                    01273000 T  0412
                    BEGIN LABEL XIT;                                                        01274000 T  0412
                                                                                  START OF SEGMENT **********    49
                       ALPHA S;                                                             01275000 T  0000
STACK(F+2) = S
                                                                                            01276000 T  0000
                       IF S ← GET(P) > 0 THEN S ← GET(GETSPACE(P));                         01277000 T  0004
                       IF S.CLASS = VARID THEN IF BOOLEAN(S,CE) THEN                        01278000 T  0006
                          IF BOOLEAN(S,TWOD) THEN                                           01279000 T  0007
                          BEGIN                                                             01280000 T  0008
                             EMITL( (T←GET(P+2),BASE),[33:7]);                              01281000 T  0011
                             EMITDESCLIT(S,ADDR);                                           01282000 T  0013
                             EMITO(LOD);                                                    01283000 T  0013
                             EMITL(T,[40:8]);                                               01284000 T  0015
                             EMITO(CDC);                                                    01285000 T  0015
                             GO TO XIT;                                                     01286000 T  0016
                          END ELSE                                                          01287000 T  0016
                          EMITNUM(GET(P+2),BASE)                                            01288000 T  0019
                          ELSE                                                              01289000 T  0019
                          IF NOT BOOLEAN(S,FORMAL) THEN IF NOT DESCREQ THEN                 01290000 T  0021
                          BEGIN                                                             01291000 T  0022
                             EMITL(S←S,ADDR);                                               01292000 T  0024
                             IF S.[37:2] = 1 THEN %REFERENCING 2ND HALF OF PRT;             01293000 T  0025
                             BEGIN                                                          01294000 T  0025
                       IF ADR ≥ 4087 THEN                                                   01295000 T  0026
                                BEGIN ADR ← ADR+1; SEGOVF END;                              01296000 T  0028
                                EMITO(XRT);                                                 01297000 T  0029
                             END;                                                           01298000 T  0029
                             GO TO XIT;                                                     01299000 T  0031
                          END;                                                              01300000 T  0031
                          EMITDESCLIT(S,ADDR);                                              01301000 T  0032
                       XIT;                                                                 01302000 T  0033
                    END EMITN;                                                    49 IS   36 LONG, NEXT SEG    6
                                                                                            01303000 T  0412
                    PROCEDURE EMITV(P); VALUE P; ALPHA P;
PRT(534) = EMITV                                                                            01304000 T  0412
                     BEGIN                                                                  01305000 T  0412
                    ALPHA S;                                                      START OF SEGMENT **********    50
STACK(F+2) = S
                                                                                            01306000 T  0000
                    IF S ← GET(P) > 0 THEN S ← GET(GETSPACE(P));                            01307000 T  0004
                    IF S.CLASS = VARID THEN                                                 01308000 T  0005
                       IF BOOLEAN(S,CE) THEN                                                01309000 T  0006
                          IF BOOLEAN(S,TWOD) THEN                                           01310000 T  0007
                             BEGIN                                                          01311000 T  0008
                                EMITL( (T←GET(P+2),BASE),[33:7]);                           01312000 T  0011
                                EMITDESCLIT(S,ADDR);                                        01313000 T  0013
                                EMITO(LOD);                                                 01314000 T  0013
                                IF S.SUBCLASS ≥ DOUBTYPE THEN                               01315000 T  0015
                                BEGIN                                                       01316000 T  0015
                                   EMITO(DUP);                                              01317000 T  0016
                                   EMITPAIR(T,[40:8]+1, COC);
```

```
                    EMITO(XCH);
                    EMITPAIR(T,[40:8], COC);
                END ELSE EMITPAIR(T,[40:8], COC);
            END
            ELSE
            IF S.SUBCLASS ≥ DOUBTYPE THEN
            BEGIN
                EMITNUM( (T_GET(P+2),BASE)+1);
                EMITOPDCLIT(S.ADDR);
                EMITNUM(T);
                EMITOPDCLIT(S.ADDR);
            END ELSE
            BEGIN
                EMITNUM(GET(P+2),BASE);
                EMITOPDCLIT(S.ADDR);
            END
        ELSE
        IF S.SUBCLASS ≥ DOUBTYPE THEN
        IF BOOLEAN(S.FORMAL) THEN
        BEGIN
            EMITDESCLIT(S.ADDR);
            EMITO(DUP);
            EMITPAIR(1, XCH);
            EMITO(INX);
            EMITO(LOD);
            EMITO(XCH);
            EMITO(LOD);
        END ELSE
        BEGIN
                    EMITOPDCLIT(S.ADDR+1);
                    EMITOPDCLIT(S.ADDR);
                END
                ELSE EMITOPDCLIT(S,ADDR)
            ELSE EMITOPDCLIT(S,ADDR);
        END EMITV;

        PROCEDURE EMITL(N); VALUE N; REAL N;
            BEGIN
            BUMPADR;
                PACK(EDOC[EDOCI],(N←0&N[36:38:10]),ADR,[46:2]);
                IF CODETOG THEN DEBUG(N);
        END EMITL;
        PROCEDURE EMITD(R, OP); VALUE R, OP; REAL R, OP;
        BEGIN
            BUMPADR;
            PACK(EDOC[EDOCI], (R ← OP & R[36:42:6]), ADR,[46:2]);
            IF CODETOG THEN DEBUG(-R);
        END EMITD;
        PROCEDURE EMITDDT(B,A,X);  VALUE B,A,X;  INTEGER B,A,X ;
PRT(535) = EMITDDT
            BEGIN % DOES DIB B, DIA A, TRB X; HANDLES [B:A:X],
            EMITD(B←B MOD 6+B DIV 6×8,DIB);  EMITD(A←A MOD 6+A DIV 6×8,DIA) ;
            EMITD(X←X,TRB) ;
            END OF EMITDDT ;
        PROCEDURE EMITPAIR(L, OP); VALUE L, OP; INTEGER L, OP;
        BEGIN
            EMITL(L);
```

```
01318000 T  0018
01319000 T  0019
01320000 T  0020
01321000 T  0022
01322000 T  0022
01323000 T  0022
01324000 T  0024
01325000 T  0024
01326000 T  0028
01327000 T  0029
01328000 T  0030
01329000 T  0031
01330000 T  0031
01331000 T  0032
01332000 T  0034
01333000 T  0035
01334000 T  0035
01335000 T  0035
01336000 T  0037
01337000 T  0038
01338000 T  0039
01339000 T  0040
01340000 T  0041
01341000 T  0042
01342000 T  0043
01343000 T  0043
01344000 T  0044
01345000 T  0045
01346000 T  0045
01347000 T  0045
01348000 T  0047
01349000 T  0048
01350000 T  0048
01351000 T  0049
01352000 T  0052
50 IS   55 LONG, NEXT SEG    6
01353000 T  0412
01354000 T  0412
01355000 T  0412
01356000 T  0414
01357000 T  0420
01358000 T  0421
01359000 T  0423
01360000 T  0423
01361000 T  0423
01362000 T  0425
01363000 T  0431
01364000 T  0432
01364010 T  0434
01364020 T  0434
01364030 T  0434
01364035 T  0441
01364040 T  0442
01365000 T  0442
01366000 T  0442
01367000 T  0442
```

```
    IF L.[37:2] = 1 THEN
    BEGIN
     IF ADR ≥ 4087 THEN
       BEGIN ADR ← ADR+1; SEGOVF END;
       EMITO(XRT);
    END;
    EMITO(OP);
END EMITPAIR;
PROCEDURE ADJUST;
         WHILE ADR.[46:2] ≠3 DO EMITO(NOP);
PROCEDURE EMITNUM(N); VALUE N; REAL N;
    BEGIN
         DEFINE CPLUS = 1792#;

         IF N.[3:6] =0 AND ABS(N) < 1024 THEN
    BEGIN
         EMITL(N);
         IF N < 0 THEN EMITO(SSN);
    END ELSE
    BEGIN
    IF ADR ≥ 4079 THEN
         BEGIN ADR ← ADR+1; SEGOVF END;
         EMITOPDCLIT(CPLUS +  (ADR+1).[46:1] + 1);
         EMITL(2);
         EMITO(GFW);
         ADJUST;
         BUMPADR;
         PACK(EDOC[EDOCI],N.[1:11],ADR.[46:2]);
         BUMPADR;
         PACK(EDOC[EDOCI],N.[12:12],ADR.[46:2]);
         BUMPADR;
         PACK(EDOC[EDOCI],N.[24:12],ADR.[46:2]);
         BUMPADR;
         PACK(EDOC[EDOCI],N.[36:12],ADR.[46:2]);
         IF N.[36:12]=49 THEN %%% IF C-REL CONSTANT LOOKS LIKE AN XRT
         EMITO(NOP) ;          %%% THEN INSURE AGAINST P-BIT INTERRUPT.
         IF CODETOG THEN DEBUGWORD(N);
    END;
END EMITNUM;

PROCEDURE EMITNUM2(HI,LO);VALUE HI,LO; REAL HI,LO;
PRT(536) = EMITNUM2
    BEGIN
    BOOLEAN B;  REAL I,N;

STACK(F+2) = B
STACK(F+3) = I
STACK(F+4) = N
    LABEL Z,X ;
         DEFINE CPLUS = 1792#;
    IF HI=0 OR LO=0 THEN BEGIN EMITNUM(LO); EMITNUM(HI); GO Z END ;
         ADJUST;
    IF ADR ≥ 4077 THEN
         BEGIN ADR←ADR+1; SEGOVF; ADR←=1 END ;
         EMITOPDCLIT(CPLUS + 2);   EMITOPDCLIT(CPLUS + 1);
         EMITPAIR(3,GFW);
    X:   FOR I ← 0 STEP 1 UNTIL 3 DO
```

```
01368000 T  0443
01369000 T  0445
01370000 T  0445
01371000 T  0446
01372000 T  0448
01373000 T  0449
01374000 T  0449
01375000 T  0450
01376000 T  0452
01377000 T  0452
01378000 T  0455
01379000 T  0455
01380000 T  0455
START OF SEGMENT **********   51
01381000 T  0000
01382000 T  0002
01383000 T  0003
01384000 T  0003
01385000 T  0005
01386000 T  0005
01387000 T  0008
01388000 T  0008
01389000 T  0011
01390000 T  0013
01391000 T  0014
01392000 T  0015
01393000 T  0015
01394000 T  0018
01395000 T  0022
01396000 T  0025
01397000 T  0029
01398000 T  0032
01399000 T  0036
01400000 T  0039
01400400 T  0043
01400600 T  0045
01401000 T  0046
01402000 T  0047
01403000 T  0047
51 IS   51 LONG, NEXT SEG    6
01404000 T  0455
01405000 T  0455
01406000 T  0455
START OF SEGMENT **********   52
01407000 T  0000
01408000 T  0000
01408100 T  0000
01409000 T  0004
01410000 T  0004
01411000 T  0005
01412000 T  0008
01413000 T  0011
01415000 T  0012
```

```
                    BEGIN
                        CASE I OF BEGIN                                                                    01416000 T  0016
                                                                                                           01417000 T  0016
PRT(537) = *CASE STATEMENT DESCRIPTOR*
                                    N ← HI.[ 1:11];                                                        01418000 T  0016
                                    N ← HI.[12:12];                                                        01419000 T  0018
                                    N ← HI.[24:12];                                                        01420000 T  0020
                                    N ← HI.[36:12];                                                        01421000 T  0022
                                END CASE;                                                                  01422000 T  0023
                                                                                                  START OF SEGMENT **********   53
                                                                                                    53 IS    5 LONG, NEXT SEG   52
                        BUMPADR; PACK(EDOC[EDOCI],N,ADR,[46:2]);                                           01423000 T  0024
                    END;                                                                                   01424000 T  0030
                        IF CODETOG THEN DEBUGWORD(HI);                                                     01425000 T  0033
                        IF NOT B THEN BEGIN B←TRUE; HI←LO; GO TO X; END;                                   01426000 T  0034
                        IF N=49 THEN    %%% IF C=REL CONSTANT LOOKS LIKE AN XRT                             01426400 T  0039
                        EMITO(NOP) ;    %%% THEN INSURE AGAINST P-BIT INTERRUPT.                           01426600 T  0039
                Z:                                                                                         01426650 T  0041
                    END EMITNUM2;                                                                          01427000 T  0041
                                                                                                    52 IS   44 LONG, NEXT SEG    6
                PROCEDURE EMITLINK(N); VALUE N; REAL N; % EMITS LINKS                                      01428000 T  0455
PRT(540) = EMITLINK
                    BEGIN
                        FORMAT FF(X35,*(33(":")),A4,":",A1,"  LINK",X10,A4,"******") ;                     01429000 T  0455
                                                                                                           01430000 T  0455
                                                                                                  START OF SEGMENT **********   54
                                                                                                  START OF SEGMENT **********   55
PRT(541) = FF
                                                                                                    55 IS   16 LONG, NEXT SEG   54
                        BUMPADR;                                                                           01431000 T  0000
                        PACK(EDOC[EDOCI],N,ADR,[46:2]);                                                    01432000 T  0002
                        IF CODETOG THEN                                                                    01433000 T  0006
                            WRITALIST(FF,4,IF ADR≤DEBUGADR THEN 1 ELSE -1,B2D(ADR,[36:10]),                01434000 T  0006
                                    B2D(ADR,[46:2]),B2D(N),0,0,0,0) ;                                      01435000 T  0012
                        IF DEBUGADR<ADR THEN DEBUGADR←ADR ;                                                01435010 T  0015
                     END EMITLINK;                                                                         01436000 T  0017
                                                                                                    54 IS   19 LONG, NEXT SEG    6
                PROCEDURE EMITSTORE(IX, EXP); VALUE IX, EXP; REAL IX, EXP;                                 01437000 T  0455
PRT(542) = EMITSTORE
                BEGIN                                                                                      01438000 T  0455
                   REAL E, VT;                                                                             01439000 T  0455
                                                                                                  START OF SEGMENT **********   56
STACK(F+2) = E
STACK(F+3) = VT
                    P ← REAL(ERRORTOG);                                                                    01440000 T  0000
                    ERRORTOG ← FALSE;                                                                      01441000 T  0000
                    E ← GET(GETSPACE(IX));                                                                 01442000 T  0001
                    IF NOT( (VT ← E.SUBCLASS) = LOGTYPE EQV EXP = LOGTYPE) OR                              01443000 T  0003
                       NOT(VT = COMPTYPE EQV EXP = COMPTYPE) THEN                                          01444000 T  0006
                       BEGIN XTA ← GET(IX+1); FLAG(86) END;                                                01445000 T  0008
                    IF E.CLASS = VARID THEN                                                                01446000 T  0011
                    BEGIN                                                                                  01447000 T  0012
                       IF BOOLEAN(E,FORMAL) OR BOOLEAN(E,CE) THEN                                          01448000 T  0013
                       BEGIN                                                                               01449000 T  0015
                          IF VT ≤ LOGTYPE THEN                                                             01450000 T  0015
                          BEGIN                                                                            01451000 T  0016
                             IF VT = INTYPE AND EXP ≥ REALTYPE THEN EMITPAIR(1, IDV);                      01452000 T  0016
                             EMITN(IX);                                                                    01453000 T  0020
                             EMITO(IF VT = INTYPE THEN ISD ELSE STD);                                      01454000 T  0020
```

```
            IF EXP ≥ DOUBTYPE THEN EMITO(DEL);                                  01455000 T  0023
         END ELSE                                                               01456000 T  0025
         BEGIN                                                                  01457000 T  0025
            EMITN(1X);                                                          01458000 T  0026
            EMITPAIR(JUNK, STN);                                                01459000 T  0026
            EMITO(STD);                                                         01460000 T  0027
            IF EXP < DOUBTYPE THEN EMITL(0);                                    01461000 T  0028
            EMITL(1);                                                           01462000 T  0030
            EMITPAIR(JUNK, LOD);                                                01463000 T  0031
            EMITO(INX);                                                         01464000 T  0032
            EMITO(STD);                                                         01465000 T  0033
         END                                                                    01466000 T  0033
      END ELSE                                                                  01467000 T  0033
      BEGIN                                                                     01468000 T  0033
         IF VT = INTYPE AND EXP ≥ REALTYPE THEN EMITPAIR(1, IDV);               01469000 T  0034
         EMITPAIR(E.ADDR, IF VT = INTYPE THEN ISD ELSE STD);                    01470000 T  0037
         IF VT ≤ LOGTYPE THEN                                                   01471000 T  0041
            IF EXP ≥ DOUBTYPE THEN EMITO(DEL) ELSE ELSE                         01472000 T  0041
         BEGIN                                                                  01473000 T  0044
            IF EXP < DOUBTYPE THEN EMITL(0);                                    01474000 T  0045
            EMITPAIR(E.ADDR+1, STD);                                            01475000 T  0047
         END                                                                    01476000 T  0049
      END                                                                       01477000 T  0049
   END ELSE                                                                     01478000 T  0049
   IF E.CLASS = ARRAYID THEN                                                    01479000 T  0049
   BEGIN                                                                        01480000 T  0051
      IF VT ≤ LOGTYPE THEN                                                      01481000 T  0051
      BEGIN                                                                     01482000 T  0052
         IF VT = INTYPE AND EXP ≥ REALTYPE THEN EMITPAIR(1, IDV);               01483000 T  0052
         IF EXP ≥ DOUBTYPE THEN                                                 01484000 T  0056
         BEGIN                                                                  01485000 T  0056
            EMITO(XCH);                                                         01486000 T  0057
            EMITO(DEL);                                                         01487000 T  0058
         END;                                                                   01488000 T  0058
         EMITO(XCH);                                                            01489000 T  0058
         EMITO(IF VT = INTYPE THEN ISD ELSE STD);                               01490000 T  0059
      END ELSE                                                                  01491000 T  0062
      BEGIN                                                                     01492000 T  0062
         IF EXP ≥ DOUBTYPE THEN                                                 01493000 T  0062
         BEGIN                                                                  01494000 T  0063
            EMITPAIR(JUNK, STD);                                                01495000 T  0064
            EMITO(XCH);                                                         01496000 T  0065
            EMITOPDCLIT(JUNK);                                                  01497000 T  0065
            EMITO(XCH);                                                         01498000 T  0066
            EMITPAIR(JUNK, STN);                                                01499000 T  0067
            EMITO(STD);                                                         01500000 T  0068
            EMITL(1);                                                           01501000 T  0069
            EMITPAIR(JUNK, LOD);                                                01502000 T  0069
            EMITO(INX);                                                         01503000 T  0070
            EMITO(STD);                                                         01504000 T  0071
         END ELSE                                                               01505000 T  0072
         BEGIN                                                                  01506000 T  0072
            EMITO(XCH);                                                         01507000 T  0072
            EMITPAIR(JUNK, STN);                                                01508000 T  0073
            EMITO(STD);                                                         01509000 T  0074
            EMITL(0);                                                           01510000 T  0075
            EMITL(1);                                                           01511000 T  0076
```

```
                        EMITPAIR(JUNK, LOD);                                                    01512000 T  0076
                        EMITO(INX);                                                             01513000 T  0077
                        EMITO(STD);                                                             01514000 T  0078
                      END                                                                       01515000 T  0079
                    END                                                                         01516000 T  0079
                  END ELSE BEGIN XTA ← GET(IX+1); FLAG(49) END;                                 01517000 T  0079
                  ERRORTOG ← ERRORTOG OR BOOLEAN(P);                                            01518000 T  0082
                END EMITSTORE;                                                                  01519000 T  0083
                                                                                          56 IS   86 LONG, NEXT SEG    6
                PROCEDURE EMITB(A,C); VALUE A,C; REAL A; BOOLEAN C;                             01520000 T  0455
                BEGIN COMMENT  GENERATES LITC AND BRANCH FROM CURRENT ADR TO ADDRESS A.         01521000 T  0455
                IF THE BOOLEAN C IS TRUE THEN THE BRANCH IS CONDITIONAL.  BEOREF IS             01522000 T  0455
                TRUE IF THE BRANCH IS BACKWARDS;                                                01523000 T  0455
                BOOLEAN BEOREF;                                                                 01524000 T  0455
                                                                                       START OF SEGMENT **********   57
STACK(F+2) = BEOREF
                    REAL SEG;                                                                   01525000 T  0000
STACK(F+3) = SEG
                IF ADR ≥ 4086 THEN                                                              01526000 T  0000
                  BEGIN ADR ← ADR+1; SEGOVF END;                                                01527000 T  0000
                    IF A < 0 THEN                                                               01528000 T  0003
                BEGIN                                                                           01529000 T  0003
                    IF BRANCHX>LBRANCH THEN FATAL(123) ;                                        01530000 T  0004
                    BRANCHX ← BRANCHES[LAX:=BRANCHX];                                           01531000 T  0006
                    BRANCHES[LAX] ← -(ADR+1);                                                   01532000 T  0007
                    EMITLINK(0);                                                                01533000 T  0009
                    EMITO(IF C THEN BFC ELSE BFW);                                              01534000 T  0010
                    EMITO(NOP);                                                                 01534100 T  0012
                END  ELSE                                                                       01535000 T  0013
                BEGIN                                                                           01536000 T  0013
                    SEG ← A.SEGNO;                                                              01537000 T  0015
                    A ← A.LINK;                                                                 01538000 T  0016
                    BEOREF ← ADR > A;                                                           01539000 T  0017
                    IF A.[46:2] =0 THEN                                                         01540000 T  0018
                BEGIN                                                                           01541000 T  0020
                    IF SEG > 0 AND SEG ≠ NSEG THEN                                              01542000 T  0020
                       EMITOPDCLIT(PRGDESCBLDR(2, 0, A.[36:10], SEG))                           01543000 T  0022
                    ELSE                                                                        01544000 T  0024
                    EMITL(ABS((ADR+2),[36:10] - A.[36:10]));                                    01545000 T  0025
                    IF BEOREF THEN                                                              01546000 T  0028
                    EMITO( IF C THEN GBC ELSE GBW) ELSE                                         01547000 T  0029
                    EMITO( IF C THEN GFC ELSE GFW);                                             01548000 T  0031
                END ELSE                                                                        01549000 T  0034
                BEGIN                                                                           01550000 T  0034
                    EMITL(ABS(ADR + 3 - A));                                                    01551000 T  0035
                    IF BEOREF THEN                                                              01552000 T  0037
                    EMITO(IF C THEN BBC ELSE BBW) ELSE                                          01553000 T  0037
                    EMITO(IF C THEN BFC ELSE BFW);                                              01554000 T  0040
                END;                                                                            01555000 T  0042
              END;                                                                              01556000 T  0042
            END EMITB;                                                                          01557000 T  0042
                                                                                          57 IS   45 LONG, NEXT SEG    6
            PROCEDURE EMITDESCLIT(N); VALUE N; REAL N;                                          01558000 T  0455
                BEGIN                                                                           01559000 T  0455
                    IF N.[37:2] = 1 THEN                                                        01560000 T  0455
                    BEGIN                                                                       01561000 T  0457
                IF ADR ≥ 4087 THEN                                                              01562000 T  0457
```

```
                        BEGIN ADR ← ADR+1; SEGOVF END;                                  01563000 T  0458
                        EMITO(XRT);                                                     01564000 T  0460
                    END;                                                                01565000 T  0461
                    BUMPADR;                                                            01566000 T  0461
                    PACK(EDOC[EDOCI],(N←3&N[36:38:10]),ADR,[46:2]);                     01567000 T  0464
                    IF CODETOG THEN DEBUG(N);                                           01568000 T  0469
                END EMITDESCLIT;                                                        01569000 T  0471
                PROCEDURE EMITOPDCLIT(N);       VALUE N; REAL N;                        01570000 T  0474
                    BEGIN                                                               01571000 T  0474
                        IF N.[37:2] = 1 THEN                                            01572000 T  0474
                        BEGIN                                                           01573000 T  0475
                    IF ADR ≥ 4087 THEN                                                  01574000 T  0475
                        BEGIN ADR ← ADR+1; SEGOVF END;                                  01575000 T  0476
                        EMITO(XRT);                                                     01576000 T  0478
                    END;                                                                01577000 T  0479
                    BUMPADR;                                                            01578000 T  0479
                    PACK(EDOC[EDOCI],(N←2&N[36:38:10]),ADR,[46:2]);                     01579000 T  0482
                    IF CODETOG THEN DEBUG(N);                                           01580000 T  0487
                END EMITOPDCLIT;                                                        01581000 T  0489
                PROCEDURE EMITLABELDESC(N); VALUE N; ALPHA N;                           01582000 T  0492
PRT(543) = EMITLABELDESC
                BEGIN                                                                   01583000 T  0492
                    LABEL XIT;                                                          01584000 T  0492
                                                                                START OF SEGMENT **********   58
                    REAL T,B,C,D ;                                                      01585000 T  0000
STACK(F+2) = T
STACK(F+3) = B
STACK(F+4) = C
STACK(F+5) = D
                    IF N ← LBLSHFT(XTA←N) = 0 OR N = BLANKS THEN                        01586000 T  0000
                    BEGIN FLAG(135); GO TO XIT END;                                     01587000 T  0004
                    IF T ← SEARCH(N) = 0 THEN                                           01588000 T  0005
                       T ← ENTER(0 & LABELID[TOCLASS], N) ELSE                          01588100 T  0007
                       IF GET(T).CLASS ≠ LABELID THEN                                   01588200 T  0010
                       BEGIN FLAG(144); GO TO XIT END;                                  01588300 T  0012
                    IF XREF THEN ENTERX(N,0&LABELID[TOCLASS]);                          01588400 T  0014
                    IF B ←(C←GET(T+2)).BASE =0 THEN                                     01589000 T  0017
                      IF (D←GET(T)).SEGNO≠0 THEN C.BASE←B←PRGDESCBLDR(2,0,D.ADDR DIV 4,  01589010 T  0020
                      D.SEGNO) ELSE                                                     01589020 T  0026
                          BEGIN BUMPPRT; C.BASE←B←PRTS END; PUT(T+2,C);                 01590000 T  0028
                    EMITL(B); EMITO(MKS);                                               01591000 T  0036
                    EMITDESCLIT(1536);          % F+0                                  01592000 T  0037
                    EMITOPDCLIT(1537);          % F+1                                  01593000 T  0038
                    EMITV(NEED(",LABEL", INTRFUNID));                                   01594000 T  0039
                    XIT;                                                                01595000 T  0040
                END EMITLABELDESC;                                                      01596000 T  0041
                                                                                  58 IS   47 LONG, NEXT SEG    6
                COMMENT TRACEBACK, OFLOWHANGERS, AND PRTSAVER ARE                       01597000 T  0492
                PROCEDURES USED TO ACCUMULATE FORMAT AND NAMELIST ARRAYS;               01598000 T  0492
                PROCEDURE TRACEBACK(M,DEX,PRT);  VALUE M,DEX,PRT; INTEGER M,DEX,PRT;    01599000 T  0492
PRT(544) = TRACEBACK
                BEGIN INTEGER I,J; REAL C;                                              01600000 T  0492
                                                                                START OF SEGMENT **********   59
STACK(F+2) = I
STACK(F+3) = J
STACK(F+4) = C
                    IF (C←GET(M+2)).BASE ≠ 0 THEN                                       01601000 T  0000
```

```
                    BEGIN I ← ADR;    ADR ← C.BASE;                                                    01602000 T  0003
                          DO BEGIN J ← GIT(ADR);                                                       01603000 T  0005
                                   ADR ← ADR - 1;                                                      01604000 T  0007
                                   EMITL(DEX);                                                         01605000 T  0008
                                   EMITPAIR(PRT,LOD);                                                  01606000 T  0009
                              END UNTIL ADR ← J = 0;                                                   01607000 T  0010
                              ADR ← I;                                                                 01608000 T  0012
                      END;                                                                             01609000 T  0012
                    INFO[M,IR,M,IC],ADDR ← PRT; PUT(M+2,0&DEX[TOBASE]);                                01610000 T  0012
                    END TRACEBACK;                                                                     01611000 T  0019
                                                                                                 59 IS   22 LONG, NEXT SEG    6
                    PROCEDURE OFLOWHANGERS(I);   VALUE I;   INTEGER I;                                 01612000 T  0492
PRT(545) = OFLOWHANGERS
                    BEGIN INTEGER J;   LABEL XIT;                                                      01613000 T  0492
                                                                                                START OF SEGMENT *********   60
STACK(F+2) = J
                    FOR J ← 1 STEP 1 UNTIL MAXNBHANG DO                                                01614000 T  0000
                    IF FNNHANG[J] = 0 THEN   %  MAKE AN ENTRY                                          01615000 T  0001
                    BEGIN FNNHANG[J] ← I;                                                              01616000 T  0002
                          PUT(I+2,J);                                                                  01617000 T  0003
                          GO TO XIT;                                                                   01618000 T  0005
                    END;                                                                               01619000 T  0005
                    XTA ← MAXNBHANG;   %  IF WE REACH HERE WERE HURTIN                                 01620000 T  0008
                    FLAG(91);                                                                          01621000 T  0008
                    XIT;                                                                               01622000 T  0009
                    END OFLOWHANGERS;                                                                  01623000 T  0010
                                                                                                 60 IS   13 LONG, NEXT SEG    6
                    PROCEDURE PRTSAVER(M,SZ,ARY);   VALUE M,SZ;                                        01624000 T  0492
PRT(546) = PRTSAVER
                    INTEGER M,SZ;   ARRAY ARY[0];                                                      01625000 T  0492
                    BEGIN INTEGER I; REAL INFA;                                                        01626000 T  0492
                                                                                                START OF SEGMENT *********   61
STACK(F+2) = I
STACK(F+3) = INFA
                    LABEL SHOW,XIT;                                                                    01626100 T  0000
                    IF (INFA←GET(M)) < 0 THEN    % PREVIOUSLY DEFINED                                  01627000 T  0000
                    BEGIN XTA ← GET(M+1);                                                              01628000 T  0001
                          FLAG(20); GO TO XIT;                                                         01629000 T  0004
                    END;                                                                               01630000 T  0005
                    IF I ← INFA   .ADDR ≠ 0 THEN   %  PRT ASSIGNED AT OFLOW TIME                       01631000 T  0005
                    SHOW;                                                                              01631100 T  0007
                    BEGIN I ← PRGDESCBLDR(1,I,0,NXAVIL ← NXAVIL + 1);                                  01632000 T  0008
                          WRITEDATA(SZ,NXAVIL,ARY);                                                    01633000 T  0011
                          FNNHANG[GET(M+2).SIZE] ← 0;                                                  01634000 T  0012
                    END ELSE   %  ADD THIS ARRAY TO HOLD                                               01635000 T  0015
                    BEGIN IF FNNPRT=0 THEN BEGIN BUMPPRT; FNNPRT←PRTS END;                             01636000 T  0015
                          IF FNNINDEX + SZ > DUMPSIZE THEN   %  ARRAY WONT FIT                        01637000 T  0021
                          IF SZ > DUMPSIZE THEN   %  ARRAY WILL NEVER FIT                              01638000 T  0023
                          BEGIN BUMPPRT;FNNHANG[GET(M+2),SIZE]←0; TRACEBACK(M,0,I←PRTS);               01639000 T  0024
                                GO TO SHOW;                                                            01640000 T  0033
                          END ELSE   %  DUMP OUT CURRENT HOLDINGS                                      01641000 T  0033
                          BEGIN FNNPRT ← PRGDESCBLDR(1,FNNPRT,0,NXAVIL ← NXAVIL + 1);                  01642000 T  0033
                                WRITEDATA(FNNINDEX,NXAVIL,FNNHOLD);                                    01643000 T  0037
                                FNNINDEX ← 0; BUMPPRT; FNNPRT ←PRTS;                                   01644000 T  0038
                          END;                                                                         01645000 T  0044
                          FNNHANG[GET(M + 2),SIZE] ←0;                                                 01646000 T  0044
                          TRACEBACK(M,FNNINDEX,FNNPRT);                                                01647000 T  0046
```

```
                    MOVEW(ARY,FNNHOLD[FNNINDEX],SZ,[36:6],SZ);                                  01648000 T  0048
                    FNNINDEX ← FNNINDEX + SZ;                                                   01649000 T  0050
                END;                                                                            01650000 T  0051
                PUT(M,-GET(M));          %  ID NOW ASSIGNED                                    01651000 T  0051
                XIT;                                                                            01651100 T  0053
                END PRTSAVER;                                                                   01652000 T  0054
                                                                                      61 IS   57 LONG, NEXT SEG    6
                PROCEDURE SEGOVF;                                                               01653000 T  0492
                  BEGIN                                                                         01654000 T  0492
                  REAL I, T, A, J, SADR, INFC, LABPRT;                                          01655000 T  0492
                                                                                    START OF SEGMENT **********   62
STACK(F+2) = I
STACK(F+3) = T
STACK(F+4) = A
STACK(F+5) = J
STACK(F+6) = SADR
STACK(F+7) = INFC
STACK(F+10) = LABPRT
                  REAL SAVINS;                                                                  01655100 T  0000
STACK(F+11) = SAVINS
                  FOR T ← 1 STEP 1 UNTIL MAXNBHANG DO                                           01656000 T  0000
                IF J←FNNHANG[T]≠0 THEN BEGIN BUMPPRT;TRACEBACK(J,FNNHANG[T]←0,PRTS) END;        01657000 T  0001
                  SEGOVFLAG ← TRUE;                                                             01661000 T  0009
                BUMPPRT;                                                                        01662000 T  0013
                  IF PRTS.[37:2]=1 THEN BEGIN                                                   01662300 T  0016
                    PACK(EDOC[EDOCI],(T←1&XRT[36:38:10]),ADR,[46:2]);                           01662500 T  0018
                    IF CODETOG THEN DEBUG(T); ADR←ADR+1 END ;                                   01662700 T  0023
                  PACK(EDOC[EDOCI], (T←2&PRTS[36:38:10]), ADR.[46:2]);                          01663000 T  0026
                  IF CODETOG THEN DEBUG(T);                                                     01664000 T  0031
                  ADR ← ADR+1;                                                                  01665000 T  0033
PACK(EDOC[EDOCI], (T←1&BFW [36:38:10]), ADR.[46:2]);                                            01666000 T  0034
IF CODETOG THEN DEBUG(T);                                                                       01667000 T  0039
    SADR ← ADR;                                                                                 01668000 T  0041
T ← PRGDESCBLDR(2, PRTS, 0, NXAVIL+1);                                                          01669000 T  0042
FOR I ← 0 STEP 1 UNTIL SHX DO                                                                   01670000 T  0044
BEGIN T ← STACKHEAD[I];                                                                         01671000 T  0046
  WHILE T ≠ 0 DO                                                                                01672000 T  0047
  BEGIN IF (A← GET(T)).CLASS = LABELID THEN                                                     01673000 T  0048
    IF A > 0 THEN                                                                               01674000 T  0050
    BEGIN                                                                                       01675000 T  0051
      ADR ← A.ADDR;                                                                             01676000 T  0052
      IF LABPRT ← (INFC←GET(T+2)).BASE = 0 THEN                                                 01677000 T  0053
      BEGIN                                                                                     01678000 T  0057
      BUMPPRT; LABPRT←PRTS;                                                                     01679000 T  0057
        PUT(T+2, INFC & PRTS[TOBASE]);                                                          01680000 T  0062
      END;                                                                                      01681000 T  0064
      WHILE ADR ≠ 0 DO                                                                          01682000 T  0064
      BEGIN J ← GET(ADR); ADR ← ADR-1;                                                          01683000 T  0065
        SAVINS←GET(ADR+2).[36:10];                                                              01683100 T  0067
        EMITOPDCLIT(LABPRT);                                                                    01684000 T  0070
        EMITO(SAVINS);                                                                          01684100 T  0070
        ADR ← J;                                                                                01685000 T  0071
      END;                                                                                      01686000 T  0072
      INFO[T.IR,T.IC].ADDR ← 0;                                                                 01687000 T  0072
    END;                                                                                        01688000 T  0077
    T ← A.LINK;                                                                                 01689000 T  0077
  END;                                                                                          01690000 T  0078
```

```
                    END;
                    FOR I ← 0 STEP 1 UNTIL LBRANCH DO                                   01691000 T  0078
                    IF T ← = BRANCHES[I] > 0 AND T < 4096 THEN                          01692000 T  0081
                    BEGIN                                                               01693000 T  0082
                       ADR ← T-1;                                                       01694000 T  0084
                             SAVINS←GIT(ADR+2),[36:10];                                 01695000 T  0085
                    BUMPPRT; EMITOPDCLIT(PRTS);                                         01695100 T  0086
                             EMITO(SAVINS);                                             01696000 T  0088
                       BRANCHES[I] ← = (PRTS+4096);                                     01696100 T  0093
                    END;                                                                01697000 T  0094
                          SEGMENT((SADR+4) DIV 4,NSEG,FALSE,FDOC);                      01698000 T  0096
                          SEGMENTSTART;                                                 01699000 T  0098
                       EMITO(NOP); EMITO(NOP);                                          01700000 T  0101
                       SEGOVFLAG ← FALSE;                                               01701000 T  0102
                 END SEGOVF;                                                            01702000 T  0103
                                                                                        01703000 T  0105
                                                                                      62 IS  110 LONG, NEXT SEG     6
                 PROCEDURE ARRAYDEC(I); VALUE I; REAL I;                                01704000 T  0492
                     BEGIN     % DECLARES ARRAYS WHOSE INFO INDEX IS I                  01705000 T  0492
                          REAL PRT,LNK,J;                                               01706000 T  0492
                                                                                    START OF SEGMENT **********    63

STACK(F+2) = PRT
STACK(F+3) = LNK
STACK(F+4) = J
                          LABEL XIT ;                                                   01706010 T  0000
                          BOOLEAN OWNID; REAL X;                                        01706100 T  0000
STACK(F+5) = OWNID
STACK(F+6) = X
                 IF DEBUGTOG THEN FLAGROUTINE("  ARRA","YDEC  ",TRUE ) ;               01706110 T  0000
                          PRT ←  GET(I).ADDR;                                           01707000 T  0002
                               IF LNK ← GET(I+2).SIZE = 0 THEN GO TO XIT ;              01708000 T  0003
                     IF (OWNID ← PRT < 1536 AND DATAPRT ≠ 0) THEN                       01708100 T  0007
BEGIN                                                                                   01708200 T  0009
     EMITOPDCLIT(DATAPRT); EMITO(LNG);                                                  01708300 T  0009
     EMITB(-1, TRUE); X ← LAX;                                                          01708400 T  0011
END;                                                                                    01708500 T  0013
     EMITO(MKS);                                                                        01709000 T  0013
     EMITDESCLIT(PRT);    % STACK OR PRT ADDRESS                                       01710000 T  0014
     IF LNK ≤ 1023 THEN                                                                 01711000 T  0014
BEGIN                                                                                   01712000 T  0015
     IF OWNID THEN EMITL(0);        % LOWER BOUND                                      01712100 T  0016
     EMITL(LNK);     % ARRAY SIZE                                                     01713000 T  0017
     EMITL(1);       % ONE DIMENSION                                                  01714000 T  0018
END                                                                                     01715000 T  0019
     ELSE                                                                               01716000 T  0019
BEGIN                                                                                   01717000 T  0019
     J ← (LNK + 255) DIV 256;                                                           01718000 T  0023
     LNK := 256;    %INCLUDE ENTIRE ARRAY SIZE IN ESTIMATE      %512-                  01719000 P  0024
     IF OWNID THEN EMITL(0);        % FIRST LOWER BOUND                                01719100 T  0025
     EMITL(J);       % NUMBER OF ROWS                                                 01720000 T  0027
     IF OWNID THEN EMITL(0);        % SECOND LOWER BOUND                               01720100 T  0027
     EMITL(256);     % SIZE OF EACH ROW                                               01721000 T  0029
     EMITL(2);       % TWO DIMENSIONS                                                 01722000 T  0030
END;                                                                                    01723000 T  0030
     EMITL(1);       % ONE ARRAY                                                      01724000 T  0030
     EMITL(IF OWNID THEN 2 ELSE 0);  %OWN OR LOCAL                                    01725000 T  0031
     EMITOPDCLIT(5); % CALL BLOCK                                                     01726000 T  0033
     ARYSZ ← ARYSZ + J + LNK;                                                         01727000 T  0034
```

```
                    IF NOT(F2TOG OR OWNID) THEN
                BEGIN
                    F2TOG ← TRUE;
                    FMITL(1);
                    FMITPAIR(FPLUS2,STD);%          F+2←TRUE
                END;
                    IF OWNID THEN FIXB(X);
                      XIT;
            IF DEBUGTOG THEN FLAGROUTINE("  ARRA","YDEC  ",FALSE) ;
             END ARRAYDEC;

            REAL PROCEDURE SEARCH(E); VALUE E; REAL E;
            BEGIN REAL T; LABEL XIT;
STACK(F+3) = T
               T ← STACKHEADIE MOD SHX];
               WHILE T ≠ 0 DO
                  IF INFO[(T+1),IR,(T+1),IC] = E THEN GO TO XIT
                  ELSE T ← INFO[T,IR,T,IC],LINK;
               XIT: SEARCH ← T;
            END SEARCH;

            INTEGER PROCEDURE GLOBALSEARCH(F); VALUE E; REAL E;
PRT(547) = GLOBALSEARCH
            BEGIN REAL T; LABEL XIT;
STACK(F+3) = T
               T ← GLOBALSTACKHEADIE MOD GHX];
               WHILE T ≠ 0 DO
                  IF INFO[(T+1),IR,(T+1),IC] = E THEN GO TO XIT
                  ELSE T ← INFO[T,IR,T,IC],LINK;
               XIT: GLOBALSEARCH ← T;
            END GLOBALSEARCH;

            PROCEDURE PURGEINFO;
PRT(550) = PURGEINFO
            BEGIN REAL J;
STACK(F+2) = J
               FLAG(13);
               FOR J ← 0 STEP 1 UNTIL SHX DO STACKHEAD[J] ← 0;
               NEXTINFO ← 2;
               NEXTCOM ← 0;
               END;

            INTEGER PROCEDURE ENTER(W, E); VALUE W, E; ALPHA W, E;
            BEGIN REAL J;
STACK(F+3) = J
               IF GLOBALNEXTINFO ≤ NEXTINFO THEN PURGEINFO;
               W,LINK ← STACKHEAD[J ← F MOD SHX];
               STACKHEAD[J] ← ENTER ← J ← NEXTINFO;
               INFO[J,IR,J,IC] ← W;
               INFO[(J←J+1),IR,J,IC] ← E;
               INFO[(J←J+1),IR,J,IC] ← 0;
               NEXTINFO ← NEXTINFO + 3;
            END ENTER;
```

```
                01728000 T  0036
                01729000 T  0037
                01730000 T  0038
                01731000 T  0040
                01732000 T  0040
                01733000 T  0041
                01733100 T  0041
                01733105 T  0043
                01733200 T  0044
                01734000 T  0046
          63 IS   53 LONG, NEXT SEG    6
                01735000 T  0492
                01736000 T  0492
         START OF SEGMENT **********   64

                01737000 T  0000
                01738000 T  0001
                01739000 T  0003
                01740000 T  0007
                01741000 T  0011
                01742000 T  0012
          64 IS   16 LONG, NEXT SEG    6
                01743000 T  0492

                01744000 T  0492
         START OF SEGMENT **********   65

                01745000 T  0000
                01746000 T  0001
                01747000 T  0003
                01748000 T  0007
                01749000 T  0011
                01750000 T  0012
          65 IS   16 LONG, NEXT SEG    6
                01751000 T  0492

                01752000 T  0492
         START OF SEGMENT **********   66

                01753000 T  0000
                01754000 T  0000
                01755000 T  0005
                01756000 T  0006
                01757000 T  0007
          66 IS   10 LONG, NEXT SEG    6
                01758000 T  0492
                01759000 T  0492
         START OF SEGMENT **********   67

                01760000 T  0000
                01761000 T  0001
                01762000 T  0004
                01763000 T  0007
                01764000 T  0010
                01765000 T  0014
                01766000 T  0018
                01767000 T  0019
```

```
                    INTEGER PROCEDURE GLOBALENTER(W, E); VALUE W, E; ALPHA W, E;                    67 IS   22 LONG, NEXT SEG    6
PRT(551) = GLOBALENTER                                                                                01768000 T  0492
                    BEGIN REAL J;                                                                     01769000 T  0492
                                                                                                  START OF SEGMENT **********   68
STACK(F+3) = J
                       IF GLOBALNEXTINFO ≤ NEXTINFO THEN PURGEINFO;                                   01770000 T  0000
                       W.LINK ← GLOBALSTACKHEAD[J ← E MOD GHX];                                       01771000 T  0001
                       GLOBALSTACKHEAD[J] ← GLOBALENTER ← J ← GLOBALNEXTINFO;                         01772000 T  0004
                       INFO[J,IR,J,IC] ← W;                                                           01773000 T  0007
                       INFO[(J←J+1),IR,J,IC] ← E;                                                     01774000 T  0010
                       INFO[(J←J+1),IR,J,IC] ← 0;                                                     01775000 T  0014
                       GLOBALNEXTINFO ← GLOBALNEXTINFO + 3;                                           01776000 T  0018
                    END GLOBALENTER;                                                                  01777000 T  0019
                                                                                                   68 IS   22 LONG, NEXT SEG    6
                    PROCEDURE LABELBRANCH(K, C); VALUE K, C; REAL K; BOOLEAN C;                       01778000 T  0492
PRT(552) = LABELBRANCH                                                                                01779000 T  0492
                    BEGIN REAL TS,T,I,X;                                                          START OF SEGMENT **********   69

STACK(F+2) = TS
STACK(F+3) = T
STACK(F+4) = I
STACK(F+5) = X
                    DEFINE LABL = K#;                                                                 01780000 T  0000
                    COMMENT LABELBRANCH GENERATES A "LITC ,,," AND "BRANCH" FROM THE                  01781000 T  0000
                    CURRENT ADDRESS TO LABEL K.   IF THE BOOLEAN C IS TRUE                            01782000 T  0000
                    THE BRANCH IS CONDITIONAL.  IF THE LABEL HAS NOT BEEN ENCOUNTERED                 01783000 T  0000
                    THEN THE APPROPRIATE LINKAGE IS MADE;                                             01784000 T  0000
                       LABEL XIT;                                                                     01785000 T  0000
                        IF ADR ≥ 4086 THEN                                                            01786000 T  0000
                    BEGIN ADR ← ADR+1; SEGOVF END;                                                    01787000 T  0000
                    IF LABL ← LBLSHFT(XTA←LABL) ≤ 0 OR LABL = BLANKS THEN                             01788000 T  0003
                    BEGIN FLAG(135); GO TO XIT END;                                                   01789000 T  0007
                    IF T ← SEARCH(LABL) ≠ 0 THEN                                                      01790000 T  0010
                    BEGIN TS ← (I ←  GET(T)).ADDR;                                                    01791000 T  0011
                       IF I.CLASS ≠ LABELID THEN BEGIN FLAG(144); GO TO XIT END;                      01791100 T  0014
                       IF I > 0 THEN                                                                  01792000 T  0017
                          BEGIN EMITLINK(TS);                                                         01793000 T  0018
                                EMITO(IF C THEN BFC ELSE BFW);                                        01794000 T  0019
                             PUT(T,I&(ADR-1)[TOADDR]);                                                01795000 T  0021
                                EMITO(NOP);                                                           01795100 T  0024
                          END ELSE                                                                    01796000 T  0025
                          IF I.SEGNO = NSEG THEN EMITB(TS, C) ELSE                                    01799000 T  0025
                       BEGIN IF TS←(X←GET(T+2)).BASE = 0 THEN                                         01800000 T  0028
                          X.BASE ← TS ← PRGDESCBLDR(2,0,(I.ADDR).[36:10],I.SEGNO);                    01801000 T  0032
                          PUT(T+2,X);                                                                 01802000 T  0037
                          EMITOPDCLIT(TS);                                                            01803000 T  0038
                          EMITO(IF C THEN BFC ELSE BFW);                                              01804000 T  0039
                       END;                                                                           01805000 T  0041
                    END ELSE                                                                          01806000 T  0041
                       BEGIN                                                                          01807000 T  0041
                          IF ADR < 0 THEN EMITO(NOP);                                                 01808000 T  0042
                          EMITLINK(0);                                                                01809000 T  0044
                             EMITO( IF C THEN BFC ELSE BFW);                                          01810000 T  0045
                          T ← ENTER(0 & LABELID[TOCLASS] & (ADR-1)[TOADDR], LABL);                    01811000 T  0047
                          EMITO(NOP);                                                                 01811100 T  0051
                       END;                                                                           01812000 T  0052
```

```
                    IF XREF THEN ENTERX(LABL,0&LABEL1D[TOCLASS]);                                  01812100 T  0052
                    XIT:                                                                           01813000 T  0054
                FND LABELBRANCH;                                                                   01814000 T  0055
                                                                                          69 IS   58 LONG, NEXT SEG    6
                PROCEDURE DATASET;      % SCANS CONSTANTS IN BLOCK DATA                         01815000 T  0492
PRT(553) = DATASET
                    BEGIN                                                                          01816000 T  0492
                        REAL LST,CUR,LTYP,CTYP,SIZ,RPT;                                            01817000 T  0492
                                                                                        START OF SEGMENT **********   70
STACK(F+2) = LST
STACK(F+3) = CUR
STACK(F+4) = LTYP
STACK(F+5) = CTYP
STACK(F+6) = SIZ
STACK(F+7) = RPT
                        REAL CUD;                                                                  01818000 T  0000
STACK(F+10) = CUD
                        BOOLEAN SGN;                                                               01819000 T  0000
STACK(F+11) = SGN
                                                                                                   01820000 T  0000
                                                                                                   01821000 T  0000
                        DEFINE TYP = GLOBALNEXT#,                                                  01822000 T  0000
                               TYPC = 18:33:15#;                                                   01823000 T  0000
                        LABEL XIT,ERROR,DPP,SPP,CPP,COMM,S;                                        01824000 T  0000
                IF DEBUGTOG THEN FLAGROUTINE("  DATA","SET   ",TRUE ) ;                          01825000 T  0000
                        DATATOG ← TRUE; FILETOG ← TRUE;                                            01826000 T  0002
                        SCAN;                                                                      01827000 T  0004
                        LSTS ← -1; LTYP ←77;                                                       01828000 T  0005
                    S:  IF TYP = PLUS OR (SGN ← TYP = MINUS ) THEN SCAN;                           01829000 T  0006
                    IF TYP = NUM THEN                                                              01830000 T  0010
BEGIN                                                                                              01831000 T  0011
                    IF NUMTYPE = STRINGTYPE AND STRINGSIZE > 1 THEN                                01832000 T  0011
BEGIN                                                                                              01833000 T  0013
                    IF LTYP ≠ 77 THEN                                                              01834000 T  0013
                        BEGIN  % NOT FIRST ENTRY-PUSH DOWN PRIOR NUMBER                            01835000 T  0014
                            IF LSTS+2 > LSTMAX THEN                                                01836000 T  0015
                            BEGIN FLAG(127); GO TO ERROR END;                                      01837000 T  0016
                            LSTT[LSTS←LSTS+1] ← RPT&LTYP[TYPC];                                    01838000 T  0020
                            LSTT[LSTS←LSTS+1] ← LST;                                               01839000 T  0022
                            LTYP ← 77;                                                             01840000 T  0025
                        END;                                                                       01841000 T  0025
                                                                                                   01841100 T  0025
                    IF LSTS + STRINGSIZE > LSTMAX THEN                                             01842000 T  0025
                    BEGIN FLAG(127); GO TO ERROR END;                                              01843000 T  0027
                            LSTT[LSTS←LSTS+1] ← STRINGSIZE & STRINGTYPE[TYPC]                      01844000 T  0028
                                              & SIZ[3:33:15];                                      01844100 T  0031
                            MOVEW(STRINGARRAY,LSTT[LSTS←LSTS+1],                                   01845000 T  0032
                                  STRINGSIZE.[36:6],STRINGSIZE);                                   01846000 T  0034
                    LSTS ← LSTS + STRINGSIZE - 1;                                                  01846100 T  0036
                    SCAN;                                                                          01847000 T  0037
                    GO TO COMM;                                                                    01848000 T  0038
END;                                                                                               01849000 T  0038
                        % GOT NUMBER                                                               01850000 T  0038
                    IF NUMTYPE = STRINGTYPE THEN                                                   01851000 T  0038
BEGIN                                                                                              01851100 T  0039
                    FNEXT ← STRINGARRAY[0];                                                        01851200 T  0040
                    NUMTYPE ← INTYPE;                                                              01851300 T  0041
```

```
     IF SIZ = 0 THEN SIZ ← 1;                                                    01851400 T  0041
END;                                                                             01851500 T  0043
     CUR ← IF SGN THEN -FNEXT ELSE FNEXT;                                        01852000 T  0043
     CTYP ← NUMTYPE; CUD ← DBLOW;                                                01853000 T  0046
                                                                                 01854000 T  0047
     SCAN;                                                                       01855000 T  0047
     IF TYP = COMMA OR TYP = SLASH THEN                                          01856000 T  0048
BEGIN                                                                            01857000 T  0050
     IF SIZ = 0 THEN SIZ ← 1;                                                    01857100 T  0050
     IF CTYP = DOUBTYPE THEN                                                     01858000 T  0052
BEGIN                                                                            01859000 T  0053
DPP; IF LTYP ≠ 77 THEN                                                           01860000 T  0053
          BEGIN                                                                  01861000 T  0054
               IF LSTS+2 > LSTMAX THEN                                           01862000 T  0055
               BEGIN FLAG(127); GO TO ERROR END;                                 01863000 T  0056
               LSTT[LSTS←LSTS+1] ← RPT&LTYP[TYPC];                               01864000 T  0058
               LSTT[LSTS←LSTS+1] ← LST;                                          01865000 T  0061
               LTYP ← 77;                                                        01866000 T  0063
          END;                                                                   01867000 T  0064
     IF LSTS+3 > LSTMAX THEN BEGIN FLAG(127); GO TO ERROR END;                   01868000 T  0064
     LSTT[LSTS←LSTS+1] ← SIZ&DOUBTYPE[TYPC];                                     01869000 T  0067
     LSTT[LSTS←LSTS+1] ← CUR;                                                    01870000 T  0069
     LSTT[LSTS ← LSTS + 1] ← CUD;                                                01871000 T  0072
     GO TO COMM;                                                                 01872000 T  0074
END;                                                                             01873000 T  0074
          % SINGLE PRECISION                                                     01874000 T  0074
SPP;                                                                             01875000 T  0074
     IF LTYP = 77 THEN                                                           01876000 T  0075
          BEGIN                                                                  01877000 T  0075
               LST ← CUR;                                                        01878000 T  0076
               LTYP ← CTYP;                                                      01879000 T  0077
               RPT ← SIZ;                                                        01880000 T  0077
               GO TO COMM;                                                       01881000 T  0078
          END;                                                                   01882000 T  0079
     IF LTYP = CTYP THEN                                                         01883000 T  0079
     IF REAL(BOOLEAN(CUR) EQV BOOLEAN(LST)) = REAL(NOT FALSE) THEN               01883100 T  0079
          BEGIN                                                                  01884000 T  0081
               RPT ← RPT + SIZ;                                                  01885000 T  0082
               GO TO COMM;                                                       01886000 T  0083
          END;                                                                   01887000 T  0084
     IF LSTS+2 > LSTMAX THEN BEGIN FLAG(127); GO TO ERROR END;                   01888000 T  0084
     LSTT[LSTS←LSTS+1] ← RPT&LTYP[TYPC];                                         01889000 T  0087
     LSTT[LSTS←LSTS+1] ← LST;                                                    01890000 T  0089
     RPT ← SIZ;                                                                  01891000 T  0092
     LST ← CUR; LTYP ← CTYP;                                                     01892000 T  0092
     GO TO COMM;                                                                 01893000 T  0094
END;                                                                             01894000 T  0094
          % TYP ≠ COMMA - CHECK FOR *                                            01895000 T  0094
     IF TYP ≠ STAR THEN BEGIN FLAG(125); GO TO ERROR END;                        01896000 T  0094
     IF CTYP  ≠ INTYPE THEN BEGIN FLAG(113); GO TO ERROR END;                    01897000 T  0097
     IF SIZ ≠ 0 OR SIZ ← CUR ≤ 0 THEN                                            01898000 T  0099
     BEGIN FLAG(64); GO TO ERROR END;                                            01899000 T  0102
     SCAN; GO TO S;                                                              01900000 T  0103
END;                                                                             01912000 T  0104
          % TYP ≠ NUM AT LABEL S                                                 01913000 T  0104
     IF SIZ = 0 THEN SIZ ← 1;                                                    01913050 T  0104
     IF NAME = "T      " OR NAME = "F      " THEN                                01913100 T  0106
```

```
BEGIN
     CUR ← REAL(NAME = "T      ");                                                  01913200 T  0108
     CTYP ← LOGTYPE;                                                               01913300 T  0109
     SCAN; GO TO SPP;                                                              01913400 T  0110
END;                                                                               01913500 T  0111
     IF TYP ≠ LPAREN THEN BEGIN FLAG(64); GO TO ERROR END;                         01913600 T  0114
                                                                                   01914000 T  0114
CPP;      % COMPLEX                                                               01915000 T  0116
     IF LTYP ≠ 77 THEN                                                             01916000 T  0116
          BEGIN                                                                    01917000 T  0117
               IF LSTS+2 > LSTMAX THEN                                             01918000 T  0117
               BEGIN FLAG(127); GO TO ERROR END;                                   01919000 T  0118
               LSTT[LSTS←LSTS+1] ← RPT&LTYP[TYPC];                                 01920000 T  0119
               LSTT[LSTS←LSTS+1] ← LST;                                            01921000 T  0121
               LTYP ← 77;                                                          01922000 T  0124
          END;                                                                     01923000 T  0126
     SCAN;                                                                         01924000 T  0127
     IF TYP = PLUS OR (SGN←TYP=MINUS) THEN SCAN;                                   01925000 T  0127
     IF TYP ≠ NUM OR NUMTYPE > REALTYPE THEN                                       01926000 T  0127
     BEGIN FLAG(64); GO TO ERROR END;                                              01927000 T  0130
     IF LSTS+2 > LSTMAX THEN BEGIN FLAG(127); GO TO ERROR END;                     01928000 T  0132
     LSTT[LSTS←LSTS+1] ← SIZ&COMPTYPE[TYPC];                                       01929000 T  0134
     LSTT[LSTS←LSTS+1] ← IF SGN THEN -FNEXT ELSE FNEXT;                            01930000 T  0137
     SCAN;                                                                         01931000 T  0140
     IF TYP ≠ COMMA THEN BEGIN FLAG(114); GO TO ERROR END;                         01932000 T  0144
     SCAN;                                                                         01933000 T  0144
     IF TYP = PLUS OR (SGN ← TYP = MINUS) THEN SCAN;                               01934000 T  0147
     IF TYP ≠ NUM OR NUMTYPE > REALTYPE THEN                                       01935000 T  0147
     BEGIN FLAG(64); GO TO ERROR END;                                              01936000 T  0150
     LSTT[LSTS←LSTS+1]←IF SGN THEN - FNEXT ELSE FNEXT ;                            01937000 T  0152
          SCAN;                                                                    01938000 T  0154
          IF TYP ≠ RPAREN THEN BEGIN FLAG(108); GO TO ERROR END;                   01939000 T  0158
          SCAN;                                                                    01940000 T  0158
     COMM;                                                                         01941000 T  0161
          SIZ ← 0;                                                                 01942000 T  0161
          IF TYP = COMMA THEN BEGIN SCAN; GO TO S; END;                            01942100 T  0162
          IF TYP = SLASH THEN GO TO XIT;                                           01943000 T  0162
          FLAG(126);                                                               01944000 T  0165
     ERROR;                                                                        01945000 T  0166
          LSTS ← 0;                                                                01946000 T  0167
          WHILE TYP ≠ COMMA AND TYP ≠ SLASH AND TYP ≠ SEMI DO SCAN;                01947000 T  0167
          IF TYP = COMMA THEN GO TO COMM;                                          01948000 T  0167
     XIT;                                                                          01949000 T  0172
          IF LTYP ≠ 77 THEN                                                        01950000 T  0173
            BEGIN                                                                  01951000 T  0174
                 IF LSTS+2>LSTMAX THEN BEGIN FLAG(127); LSTS←0 END;                01952000 T  0174
                 LSTT[LSTS←LSTS+1] ← RPT&LTYP[TYPC];                               01953000 T  0175
                 LSTT[LSTS←LSTS+1] ← LST;                                          01954000 T  0178
            END;                                                                   01955000 T  0181
          IF LSTS+1 > LSTMAX THEN BEGIN FLAG(127); LSTS ← 0 END;                   01956000 T  0183
          LSTT[LSTS←LSTS+1] ←0;                                                    01957000 T  0183
     IF DEBUGTOG THEN FLAGROUTINE("  DATA","SET    ",FALSE) ;                      01958000 T  0186
          DATATOG ← FALSE; FILETOG ← FALSE;                                        01959000 T  0189
     END DATASET;                                                                  01960000 T  0191
                                                                                   01961000 T  0193
                                                                  70 IS  199 LONG. NEXT SEG    6
     ALPHA PROCEDURE CHECKDO;                                                      01962000 T  0492
PRT(554) = CHECKDO
```

```
                    BEGIN ALPHA X, T; INTEGER N;                                                    01963000 T  0492
                                                                                                  START OF SEGMENT **********   71
STACK(F+3) = X
STACK(F+4) = T
STACK(F+5) = N
                    STREAM PROCEDURE CKDO(A, ID, LAB);                                              01964000 T  0000

PRT(555) = CKDO
                    BEGIN                                                                           01965000 T  0000
                       SI ← A; SI ← SI+4; DI ← LAB; DI ← DI+2;                                      01966000 T  0000
                       5(IF SC ≥ "0" THEN DS ← CHR ELSE JUMP OUT);                                  01967000 T  0001
                       DI ← ID; DI ← DI+2;                                                          01968000 T  0003
                       6(IF SC = ALPHA THEN DS ← CHR ELSE JUMP OUT);                                01969000 T  0003
                    END CKDO;                                                                       01970000 T  0006
                       IF (XTA←HOLDID[0]).[12:24] = "FILE" THEN FLOG(37) ELSE                       01971000 T  0006
                           IF XTA.[12:12] ≠ "DO" THEN FLOG(17) ELSE                                 01971010 T  0010
                       BEGIN                                                                        01972000 T  0014
                          X ← T ← BLANKS;                                                           01973000 T  0016
                          CKDO(HOLDID[0], X, T);                                                    01974000 T  0017
                             IF X=BLANKS THEN FLOG(105);                                            01974500 T  0018
                          IF T ← LBLSHFT(T) < 0 OR T = BLANKS THEN FLOG(17) ELSE                    01975000 T  0020
                          TEST ← NEED(T, LABELID);                                                  01976000 T  0025
                          DOLAB[DT] ← T;                                                            01977000 T  0027
                          IF XREF THEN ENTERX(T,0&LABELID[TOCLASS]);                                01977100 T  0028
                             IF  GET(TEST) < 0 THEN   % TEST FOR PREV DEFINITION                    01978000 T  0031
                          BEGIN                                                                     01979000 T  0032
                             XTA ← GET(TEST+1);                                                     01980000 T  0033
                             FLAG(15);                                                              01981000 T  0035
                             DT ← DT-1;                                                             01982000 T  0035
                          END;                                                                      01983000 T  0037
                          IF N ← SEARCH(X) = 0 THEN                                                 01984000 T  0037
                             N←ENTER(TIPE[IF T←X.[12:6]≠"0" THEN T ELSE 12],X) ;                    01985000 T  0038
                             CHECKDO ← GETSPACE(N);                                                 01987000 T  0044
                             IF XREF THEN ENTERX(X,1&GET(N) [15:15:9]);                             01987100 T  0045
                             IF (X←GET(N)).SUBCLASS > REALTYPE OR X.CLASS ≠ VARID THEN              01988000 T  0048
                             BEGIN XTA ← GET(N+1); FLAG(84) END;                                    01989000 T  0052
                             IF GET(FX1).CLASS = UNKNOWN THEN PUT(FX1+1, ".....");                  01990000 T  0055
                       END;                                                                         01991000 T  0059
                    END CHECKDO;                                                                    01992000 T  0059
                                                                                                  71 IS   63 LONG, NEXT SEG    6
                    PROCEDURE FIXB(N); VALUE N; REAL N;                                             01993000 T  0492
                    BEGIN                                                                           01994000 T  0492
                       REAL T, U, FROM;                                                             01995000 T  0492
                                                                                                  START OF SEGMENT **********   72
STACK(F+2) = T
STACK(F+3) = U
STACK(F+4) = FROM
                       LABEL XIT, BIGJ;                                                             01996000 T  0000
                    IF DEBUGTOG THEN FLAGROUTINE("    FI","XB    ",TRUE ) ;                        01996010 T  0000
                       IF N ≥ 10000 THEN FROM ← N-10000 ELSE                                        01997000 T  0002
                       IF FROM ← - BRANCHES[N] > 4095 THEN                                          01998000 T  0004
                       BEGIN                                                                        01999000 T  0009
                          ADJUST;                                                                   02000000 T  0010
                          T ← PRGDESCBLDR(2, FROM.LINK, (ADR+1).[36:10], NSEG);                     02001000 T  0010
                          GO TO XIT;                                                                02002000 T  0014
                       END;                                                                         02003000 T  0016
                    T ← ADR;   ADR ← FROM - 1;                                                      02004000 T  0016
                    IF (T + 1).[46:2] = 0 THEN GO TO BIGJ;                                          02005000 T  0018
```

```
                IF (U ← T - 2 - ADR) ≤ 1023 THEN EMITL(U) ELSE                                  02006000 T  0020
                BEGIN ADR ← T;   ADJUST;  T ← ADR;  ADR ← FROM - 1;                             02007000 T  0023
                BIGJ; EMITL((T+1),[36:10] = (ADR+2),[36:10]);                                   02008000 T  0027
                      EMITO(IF BOOLEAN(GIT(FROM + 1),[36:1]) THEN GFW ELSE GFC);                02009000 T  0031
                END;                                                                            02010000 T  0035
                ADR ← T;                                                                        02011000 T  0035
                   XIT;                                                                         02012000 T  0035
                   IF N < 10000 THEN BEGIN                                                      02013000 T  0036
                   BRANCHES[N] ← BRANCHX;                                                       02014000 T  0037
                   BRANCHX ← N;                                                                 02015000 T  0038
                   END;                                                                         02016000 T  0039
                IF DEBUGTOG THEN FLAGROUTINE("    F1","XB    ",FALSE) ;                        02016010 T  0039
                END FIXB;                                                                       02017000 T  0041
                                                                                          72 IS   47 LONG, NEXT SEG    6
                PROCEDURE  DATIME;          % PRODUCES HEADING LINE FOR LISTING,               02018000 T  0492
                    BEGIN                                                                       02019000 T  0492
                    INTEGER D ;                                                                 02020000 T  0492
                                                                                         START OF SEGMENT **********   73
STACK(F+2) = D
                   FORMAT T(X9,"B 5 7 0 0    F O R T R A N    C O M P I L A T I O N   ",       02021000 T  0000
                                                                                         START OF SEGMENT **********   74
PRT(556) = T
                "XV1,0"                                                                         02022000 T  0000
                         ,",",A2,"-",A8,"DAY, ",2(A2,"/"),A2,",",A2,":",A2," H,"/);             02022500 T  0000
                                                                                          74 IS   30 LONG, NEXT SEG   73
                    WRITALIST(T,7,                                                              02024000 T  0000
                     "16"                                                          %999-        02025000 P  0001
                           ,TIME(6),(D←TIME(5)),[12:12],D,[24:12],D,[36:12],                    02026000 T  0001
                          D←(D←RTI DIV 216000) MOD 10+D DIV 10×64,                              02027000 T  0005
                          D←(D←RTI DIV 3600 MOD 60) MOD 10+D DIV 10×64,0) ;                     02028000 T  0009
                     IF D←LINE.TYPE=10 OR D=12 OR D=13 THEN                                     02029000 T  0014
                          BEGIN                                                                 02030000 T  0019
                          LOCK(LINE);  LINE.AREAS←0;  LINE.AREASIZE←0 ;                         02031000 T  0020
                          IF D≠12 THEN LINE.TYPE←12;  SPACE(LINE,2) ;                           02032000 T  0027
PRT(557) = INPUT(W)
                          END ;                                                                 02033000 T  0034
                     FIRSTCALL←FALSE ;                                                          02034000 T  0034
                     END DATIME;                                                                02039000 T  0036
                                                                                          73 IS   44 LONG, NEXT SEG    6
                PROCEDURE PRINTCARD;                                                            02040000 T  0492
                BEGIN                                                                           02041000 T  0492
                   STREAM PROCEDURE MOVE(P, Q, A); VALUE A, Q;                                  02042000 T  0492
                                                                                         START OF SEGMENT **********   75
PRT(560) = MOVE
                   BEGIN                                                                        02043000 T  0000
                      S1 ← Q; DI ← P;                                                           02044000 T  0000
                      DS ← CHR;                                                                 02045000 T  0000
                      D1 ← DI+11; SI ← LOC A;                                                   02046000 T  0000
                      DS ← 4 DEC;                                                               02047000 T  0001
                   END MOVE;                                                                    02048000 T  0001
                   STREAM PROCEDURE MOVEBACK(P);                                                02049000 T  0001
PRT(561) = MOVEBACK
                   BEGIN DI ← P; DS ← LIT "]" END;                                              02050000 T  0001
                    MOVE (CRD[9],BUFL,(ADR+1),[36:10]);                                         02051000 T  0003
                   IF FIRSTCALL THEN DATIME;                                                    02052000 T  0005
                   IF UNPRINTED THEN WRITAROW(15,CRD) ;                                         02053000 T  0007
                    MOVEBACK(CRD[9]);                                                           02054000 T  0010
```

```
                    IF SEQERRORS THEN WRITAROW(14,ERRORBUFF) ;
                END PRINTCARD;

                BOOLEAN PROCEDURE READACARD; FORWARD;
PRT(562) = READACARD
                PROCEDURE FILEOPTION;  FORWARD;
PRT(563) = FILEOPTION
                BOOLEAN PROCEDURE LABELR;
PRT(564) = LABELR
                BEGIN

                LABEL XIT, LOOP;

                BOOLEAN STREAM PROCEDURE CHECK(CD, LAB);
PRT(565) = CHECK
                BEGIN LABEL XIT; LOCAL T1;
                   SI ← CD;
                   IF SC ≠ " " THEN IF SC < "0" THEN
                         BEGIN DI ← LAB;   DI ← DI + 2;
                               DS ← 6 CHR;  GO TO XIT;
                         END;
                   DI ←LOC T1; DS ← 6 LIT " "; DI ← DI = 6;
                   5(IF SC ≥ "0" THEN DS ← CHR ELSE SI ← SI+1);
                   DI ←LAB; DI ←DI + 2; SI ← LOC T1;
                    5(IF SC ≠ "0" THEN JUMP OUT; SI ← SI + 1);
                   5(IF SC ≥ "0" THEN DS ← CHR ELSE JUMP OUT);
                   TALLY ← 1;
                   XIT: CHECK ← TALLY;
                END CHECK;
                BOOLEAN STREAM PROCEDURE BLANKCARD(CD);
PRT(566) = BLANKCARD
                   BEGIN  LABEL XIT;
                      SI ← CD;
                      2(36( IF SC ≠ " " THEN JUMP OUT 2 TO XIT ELSE SI ← SI + 1));
                      TALLY ← 1;
                      XIT: BLANKCARD ← TALLY;
                   END BLANKCARD;
                      LOOP:
                      LABL ← BLANKS;
                      IF LABELR ← NOT READACARD THEN GO TO XIT;
                         IF NOT CHECK(CRD[0],LABL) THEN
                      BEGIN IF LABL = "FILE  " THEN FILEOPTION ELSE
                         BEGIN IF LISTOG THEN PRINTCARD;
                            IF (XTA ← LABL).[12:6] ≠ "C" THEN FLAG(135);
                         END;
                         GO TO LOOP;
                      END;
                      IF ENDSEGTOG THEN IF BLANKCARD(CRD) THEN
                         BEGIN IF LISTOG THEN PRINTCARD;  GO TO LOOP END ELSE
                       BEGIN SEGMENTSTART;
                         IF LISTOG THEN PRINTCARD;
                         IF LABL = BLANKS THEN GO TO XIT;
                      END ELSE
                      BEGIN
                            IF LABL = BLANKS THEN
                             BEGIN IF LISTOG THEN PRINTCARD; GO TO XIT END;
                            IF ADR > 0 THEN ADJUST;
```

```
                02054010 T  0011
                02055000 T  0013
          75 IS   14 LONG, NEXT SEG    6
                02056000 T  0492

                02058000 T  0492

                02059000 T  0492

                02060000 T  0492
                02061000 T  0492
                02062000 T  0492
          START OF SEGMENT **********   76
                02063000 T  0000

                02064000 T  0000
                02065000 T  0000
                02066000 T  0000
                02067000 T  0001
                02068000 T  0002
                02069000 T  0003
                02070000 T  0003
                02071000 T  0004
                02072000 T  0006
                02073000 T  0007
                02074000 T  0009
                02075000 T  0011
                02076000 T  0011
                02077000 T  0012
                02077100 T  0013
                02077200 T  0013
                02077300 T  0014
                02077400 T  0014
                02077600 T  0017
                02077700 T  0017
                02077800 T  0018
                02078000 T  0019
                02079000 T  0020
                02080000 T  0020
                02081000 T  0022
                02082000 T  0025
                02083000 T  0027
                02083100 T  0030
                02083200 T  0033
                02084000 T  0033
                02085000 T  0034
                02086000 T  0034
                02086500 T  0037
                02087000 T  0039
                02088000 T  0040
                02089000 T  0042
                02090000 T  0043
                02091000 T  0043
                02092000 T  0044
                02093000 T  0045
                02094000 T  0047
```

```
            IF LISTOG THEN PRINTCARD;                                                    02095000 T  0049
      END;                                                                               02096000 T  0051
      XIT;                                                                               02116000 T  0051
END LABELR;                                                                              02117000 T  0052
                                                                          76 IS   55 LONG, NEXT SEG     6
PROCEDURE FILEOPTION;                                                                    02118000 T  0492
BEGIN  COMMENT  THIS PROCEDURE PROCESSES THE OPTIONAL FILE CONTROL CARD.                  02119000 T  0492
THE WORD "FILE" APPEARS IN COL. 1 - 4.  COL. 5 AND 6 ARE BLANK.                          02120000 T  0492
#1 BELOW IS REQUIRED. OTHER ENTRIES MAY BE AS SPARSE AS DESIRED.                         02121000 T  0492
1. FILE  <NUM> = <MULTI-FILE ID> / <FILE ID>                                             02122000 T  0492
                OR                                                                       02123000 T  0492
   FILE  <NUM> = <FILE ID>                                                               02124000 T  0492
      THE FOLLOWING "/" IS A DOCUMMENTARY OR.                                            02125000 T  0492
      THE SEQUENCE OF RESERVED WORDS MUST BE MAINTAINED.                                 02126000 T  0492
2. UNIT=PRINT/READER/PUNCH/DISK/TAPE7/TAPE9/REMOTE   (UNIT DESIGNATE).                   02127000 T  0492
3. UNLABELED     (FOR UNLABELED TAPES)                                                   02128000 T  0492
4. ALPHA      (FOR ALPHA RECORDING MODE)                                                 02129000 T  0492
5. BCL      (IGNORED,     FOR 3500 USE)                                                  02130000 T  0492
6. FIXED    (IGNORED,     FOR 3500 USE)                                                  02131000 T  0492
7. SAVE = <NUM>      (SAVE FACTOR IN DAYS)                                               02132000 T  0492
8. LOCK      (LOCK FILE AT EOJ)                                                          02133000 T  0492
9. RANDOM/SERIAL/UPDATE      (DISK USE)                                                  02134000 T  0492
10. AREA = <NUM>      (DISK RECORDS/ROW)                                                 02135000 T  0492
11. BLOCKING = <NUM>      (RECORD PER BLOCK)                                             02136000 T  0492
12. RECORD = <NUM>      (RECORD SIZE)                                                    02137000 T  0492
13. BUFFER = <NUM>      (# OF BUFFERS)                                                   02138000 T  0492
14. WORKAREA      (IGNORED, FOR 3500)   ;                                                02139000 T  0492
ALPHA P,KEEP;   BOOLEAN TOG,CA,TS;   LABEL XIT;                                          02140000 T  0492
                                                                        START OF SEGMENT **********   77
STACK(F+2) = P
STACK(F+3) = KEEP
STACK(F+4) = TOG
STACK(F+5) = CA
STACK(F+6) = TS
            COMMENT   INXFIL = INFO.ADINFO,  MULTI FILE ID = FILEINFO[1,INXFIL],         02141000 T  0000
                  FILE ID = FILEINFO[2,INXFIL],  DISK RECORDS = FILEINFO[3,INXFIL],      02142000 T  0000
                  FILEINFO[0,INXFIL] FROM RIGHT TO LEFT IS;                              02143000 T  0000
            INTEGER  %  NAME          USE                          BITS                  02144000 T  0000
                        BUFF,         %  # BUFFERS                   6                   02145000 T  0000
STACK(F+7) = BUFF
                        RECORD,       %  RECORD SIZE                 12                  02146000 T  0000
STACK(F+10) = RECORD
                        BLOCK,        %  BLOCK SIZE                  12                  02147000 T  0000
STACK(F+11) = BLOCK
                        SAVER,        %  SAVE FACTOR                 12                  02148000 T  0000
STACK(F+12) = SAVER
                        SPIN,         %  REW & LOCK @ EOJ            2                   02149000 T  0000
STACK(F+13) = SPIN
                        ALPH;         %  RECORDING MODE              1                   02150000 T  0000
STACK(F+14) = ALPH
            PROCEDURE FETCH;                                                             02151000 T  0000
PRT(567) = FETCH
            BEGIN SCAN;  XTA ← SYMBOL;                                                   02152000 T  0000
                  IF NEXT=COMMA OR NEXT=MINUS THEN                                       02153000 T  0001
                  BEGIN SCAN;  XTA ← SYMBOL;                                             02154000 T  0003
                        IF NEXT ≠ ID THEN FLOG(37);                                      02155000 T  0004
                  END;                                                                   02156000 T  0006
```

```
                FND FETCH;                                                                                  02157000 T  0006
                INTEGER STREAM PROCEDURE MAKEINT(XTA);                                                      02158000 T  0007
PRT(570) = MAKEINT
                BEGIN LABEL LOOP;  LOCAL T;                                                                 02159000 T  0007
                SI ← XTA;  SI ← SI + 2;                                                                     02160000 T  0007
                LOOP;  IF SC ≥ "0" THEN                                                                     02161000 T  0007
                     BEGIN TALLY ← TALLY + 1;  SI ← SI + 1;  GO TO LOOP END;                                02162000 T  0008
                T ← TALLY;  SI ← XTA;  SI ← SI + 2;  DI ← LOC MAKEINT;  DS ← T OCT;                         02163000 T  0009
                END MAKEINT;                                                                                02164000 T  0011
                INTEGER PROCEDURE REPLACEMENT;                                                              02165000 T  0012
PRT(571) = REPLACEMENT
                BEGIN                                                                                       02166000 T  0012
                FETCH;   IF NEXT = EQUAL THEN                                                               02167000 T  0012
                BEGIN FETCH;  IF XTA.[12:6] ≤ 11 THEN BEGIN REPLACEMENT ← MAKEINT(XTA);                     02168000 T  0014
                     FETCH END ELSE FLOG(37);                                                               02169000 T  0019
                END ELSE FLOG(37);                                                                          02170000 T  0020
                END REPLACEMENT;                                                                            02171000 T  0022
                INTEGER STREAM PROCEDURE SRI7(S);                                                           02172000 T  0024
PRT(572) = SRI7
                BEGIN SI ← S;  DI ← LOC SRI7;                                                               02173000 T  0024
                     SI ← SI + 2;  DI ← DI + 1;     DS ← 7 CHR;                                             02174000 T  0025
                END SRI7;                                                                                   02175000 T  0026
                COMMENT * * * * * START OF CODE * * * * ;                                                   02176000 T  0027
                IF DEBUGTOG THEN FLAGROUTINE(" FILEO","PTION ",TRUE ) ;                                     02176010 T  0027
                ERRORTOG ← FALSE;                                                                           02176100 T  0030
                IF LISTOG THEN PRINTCARD;                                                                   02177000 T  0030
                XTA ← "FILE  ";                                                                             02178000 T  0032
                IF NSEG ≠ 0 THEN FLAG(60);                                                                  02179000 T  0033
                IF INXFIL ← INXFIL + 1 > MAXOPFILES THEN                                                    02180000 T  0035
                BEGIN FLAG(59); GO TO XIT END;                                                              02181000 T  0037
BUMPPRT;                                                                                                    02182000 T  0042
MAXFILES ← MAXFILES + 1;                                                                                    02183000 T  0045
FILETOG ← TRUE;  SCN ← 1;   %  START SCAN MAINTAINENCE                                                      02184000 T  0047
FETCH;  IF XTA.[12:6] > 11 THEN BEGIN FLAG(37);  GO TO XIT END;                                             02185000 T  0048
     IF XTA.[12:6] = 0 THEN BEGIN FLOG(037);  GO TO XIT END ;                                               02185010 T  0052
IF T ← GLOBALSEARCH(P ← 0&","[12:42:6]&XTA[18:12:30]) ≠ 0 THEN FLAG(20)                                     02186000 T  0055
ELSE BEGIN P← GLOBALENTER(-0&PRTS[TOADDR]&FILEID[TOCLASS],P);                                               02187000 T  0060
          PUT(P+2,GET(P+2)&INXFIL[TOADINFO]);                                                               02188000 T  0064
          IF XREF THEN ENTERX(XTA &1[TOCE],1&FILEID[TOCLASS]);                                              02188100 T  0068
     END;                                                                                                   02189000 T  0072
INFO ← GET(P + 2);                                                                                          02190000 T  0072
FETCH;   IF NEXT = EQUAL THEN FETCH ELSE                                                                    02191000 T  0073
   BEGIN FLOG(37);  GO TO XIT;  END;                                                                        02192000 T  0076
IF NEXT = SEMI THEN                                                                                         02193000 T  0077
BEGIN FLAG(37); GO TO XIT END                                                                               02194000 T  0078
    ELSE FILEINFO[2,INXFIL] ← SRI7(ACCUM[1]);                                                               02195000 T  0080
FETCH;   IF NEXT = SLASH THEN                                                                               02196000 T  0084
     BEGIN FILEINFO[1,INXFIL] ← FILEINFO[2,INXFIL];   %  MULTI FILE ID                                       02197000 T  0085
          FETCH;                                                                                            02198000 T  0088
          IF NEXT = SEMI THEN BEGIN FLAG(37);   GO TO XIT;  END                                             02199000 T  0089
               ELSE FILEINFO[2,INXFIL] ← SRI7(ACCUM[1]);                                                    02200000 T  0091
          FETCH;                                                                                            02201000 T  0095
     END;                                                                                                   02202000 T  0096
IF XTA = "UNIT  " THEN                                                                                      02203000 T  0096
BEGIN                                                                                                       02204000 T  0096
   FETCH;                                                                                                   02205000 T  0097
  IF NEXT = EQUAL THEN FETCH ELSE FLOG(37);                                                                 02206000 T  0097
```

```
          INFC.LINK ← KEEP ← (IF TOG ← XTA = "PRINT "                                  02207000 T  0101
                                 OR XTA = "PRINTE" THEN 18 ELSE                        02208000 T  0102
                              IF CA ← TOG ← XTA = "READ  "                             02209000 T  0105
                                 OR XTA = "READER" THEN 2 ELSE                         02210000 T  0105
                              IF CA ← TOG ← XTA = "PUNCH " THEN 0 ELSE                 02211000 T  0109
                                 IF TOG ← XTA = "DISK  " THEN 12 ELSE                  02212000 T  0112
                                 IF TOG ← XTA = "TAPE  "                               02213000 T  0115
                                    OR XTA = "TAPE7 " THEN 2 ELSE                      02214000 T  0115
                                   IF TOG ← XTA = "PAPER " THEN 8 ELSE                 02214100 C  0118
                              IF CA ←TOG←XTA= "REMOTE" THEN 19 ELSE                    02214500 T  0121
                                 IF TOG ← XTA = "TAPE9 " THEN 2 ELSE 2);               02215000 T  0124
       IF TOG THEN FETCH ELSE FLOG(37)                                                 02216000 T  0128
END ELSE INFC.LINK←KEEP←IF DCINPUT THEN 12 ELSE 2 ;                                    02217000 T  0142
TS←KEEP=12 ;                                                                           02217050 T  0147
IF XTA="BACKUP" THEN                                                                   02217100 T  0148
    BEGIN                                                                              02217150 T  0149
    FETCH;  IF KEEP≠0 AND KEEP≠18 THEN FLAG(37) ;                                      02217200 T  0150
    IF TOG←XTA="DISK  " THEN KEEP←IF KEEP=0 THEN 22 ELSE 15                            02217250 T  0153
    ELSE IF TOG←XTA="TAPE  " THEN KEEP←IF KEEP=0 THEN 20 ELSE 6                        02217275 T  0157
         ELSE BEGIN                                                                    02217300 T  0164
              TOG←XTA="ALTERN";  KEEP←IF KEEP=0 THEN 25 ELSE 16 ;                      02217350 T  0167
              END ;                                                                    02217400 T  0171
    IF TOG THEN FETCH;  INFC.LINK←KEEP ;                                               02217450 T  0171
    END ;                                                                              02217500 T  0174
IF XTA = "UNLABE" THEN  %   FOR UNLABELED TAPES                                        02218000 T  0174
     BEGIN IF KEEP = 2 THEN INFC   .LINK ← 9; FETCH;  END;                             02219000 T  0174
IF XTA = "ALPHA " THEN FETCH ELSE IF KEEP = 2 THEN ALPH ← 1;   %  MODE                 02220000 T  0178
IF XTA = "BCL   " THEN FETCH;   %  FOR  B3500                                         02221000 T  0186
IF XTA = "FIXED " THEN FETCH;   %  FOR  B3500                                         02222000 T  0187
IF XTA = "SAVE  " THEN SAVER ← REPLACEMENT;                                            02223000 T  0189
IF XTA = "LOCK  " THEN BEGIN SPIN ← 2; FETCH END;  %  REW & LOCK AT EOJ                02224000 T  0191
     IF TOG ← XTA = "RANDOM" THEN T ← 10 ELSE                                          02225000 T  0194
     IF TOG ← XTA = "SERIAL" THEN T ← 12 ELSE                                          02226000 T  0196
     IF TOG ← XTA = "UPDATE" THEN T ← 13;                                              02227000 T  0205
IF TOG THEN                                                                            02228000 T  0209
    BEGIN IF KEEP=12 THEN INFC.LINK←T ELSE FLAG(37); FETCH END ;                       02229000 T  0209
IF XTA="AREA  " THEN                                                                   02230000 T  0216
    BEGIN                                                                              02230010 T  0217
    IF KEEP≠12 THEN     FLAG(37);                                                      02230020 T  0217
    T←REPLACEMENT;                                                                     02230030 T  0219
    IF XTA="EU    " THEN                                                               02230040 T  0220
         IF I←REPLACEMENT>19 THEN FLAG(37)                                             02230045 T  0221
         ELSE T.EUNF←I+1;%              0 MEANS EU NOT SPECIFIED                       02230050 T  0224
    IF XTA="SPEED " THEN                                                               02230060 T  0229
         BEGIN                                                                         02230070 T  0230
         FETCH;                                                                        02230080 T  0230
         IF NEXT=EQUAL THEN                                                            02230090 T  0231
              BEGIN                                                                    02230100 T  0231
              FETCH;                                                                   02230110 T  0232
              IF XTA.[12:6]≤SLOWV THEN                                                 02230115 T  0232
                   IF I←MAKEINT(XTA)>SLOWV THEN FLAG(37)                               02230120 T  0234
                   ELSE                                                                02230125 T  0237
              ELSE IF XTA="FAST  " THEN          T.SPDF←FASTV                         02230130 T  0238
                   ELSE IF XTA="SLOW  " THEN     T.SPDF←SLOWV                         02230140 T  0242
                        ELSE FLOG(37);                                                 02230150 T  0246
              FETCH;                                                                   02230160 T  0250
              END                                                                      02230170 T  0251
```

```
            ELSE FLOG(37);
            END;
        IF XTA="SENSIT" THEN
            BEGIN
            T.SENSF:=1;
            FETCH;
            END;
        FILEINFO[3,INXFIL]:=T;
        END;
    IF XTA = "BLOCKI" THEN BLOCK ← REPLACEMENT & 1[2:47:1];
RECORD ← IF XTA="RECORD" THEN REPLACEMENT ELSE IF CA THEN 10 ELSE IF TS
        AND NOT (BOOLEAN(BLOCK,[2:1])) THEN 10 & 1[2:47:1] ELSE 17;
BUFF   ← IF XTA = "BUFFER" THEN REPLACEMENT ELSE 2;
    IF XTA = "WORKAR" THEN FETCH;    %  INGNORED, FOR 3500
    IF BUFF<1 OR BUFF>32 THEN BEGIN XTA←"BUFFER"; FLAG(152) END ;
    IF RECORD<1 THEN BEGIN XTA←"RECORD"; FLAG(152)END ;
    IF SAVER>999 THEN BEGIN XTA←"SAVE  "; FLAG(152) END ;
    IF T←INFC.LINK=10 OR T=12 OR T=13 THEN %%% ARE IN DISK FILE
        BEGIN
        IF BOOLEAN(RECORD,[2:1])THEN BLOCK ← 300
        ELSE
            IF BLOCK←BLOCK×RECORD>1890 OR RECORD>1023 THEN
                BEGIN XTA←"BK/REC"; FLAG(152)  END
        END
    ELSE IF BLOCK←BLOCK×RECORD>1023 OR RECORD>1023 THEN IF KEEP ≠ 2 THEN
        BEGIN XTA←"BK/REC"; FLAG(58 ) END ELSE BEGIN RECORD←257; BLOCK←0END;
    FILEINFO[0,INXFIL] ← 0&BUFF[42:42:6]&RECORD[30:36:12]&BLOCK[18:36:12]
          &SAVER[6:36:12]&SPIN[4:46:2]&ALPH[3:47:1];
XIT:    IF NEXT ≠ SEMI THEN
          BEGIN FLOG(37);  DO SCAN UNTIL NEXT = SEMI;  END;
        FILETOG ← FALSE;    %  END SCAN MAINTAINENCE
        PUT(P+2,INFC);
    IF DEBUGTOG THEN FLAGROUTINE(" FILEO","PTION ",FALSE) ;
    END FILEOPTION;

    PROCEDURE DOLOPT;
PRT(573) = DOLOPT
    BEGIN
    REAL STREAM PROCEDURE SCAN(BUF,ID); VALUE BUF;

PRT(574) = SCAN
        BEGIN     LABEL     LP,LA,LE,XIT;
                  SI ← BUF; DI ← ID; DS ← 2 LIT "0";
            LP:   IF SC = " " THEN BEGIN SI←SI+1; GO TO LP; END;
                  IF SC = "," THEN BEGIN SI←SI+1; GO TO LP; END;
                  IF SC = "+" THEN BEGIN DS←CHR; GO TO XIT; END;
                  IF SC = "-" THEN BEGIN DS←CHR; GO TO XIT; END;
                  IF SC < "A" THEN BEGIN DI ← ID; DS ← 8 LIT "+0000001";
        LE:                              SI←SI+1;
                                         GO TO XIT;
                                 END;
                  IF SC="×" THEN    GO TO LE;%      THIS IS > "A"
                  IF SC="≠" THEN    GO TO LE;%      THIS IS > "A"
                  6(IF SC = ALPHA THEN DS ← CHR ELSE JUMP OUT);
            LA:   IF SC = ALPHA THEN BEGIN SI←SI+1; GO TO LA; END;
                  XIT:
                  SCAN ← SI;
```

```
02230180 T  0251
02230190 T  0252
02230200 T  0252
02230210 T  0253
02230220 T  0253
02230230 T  0255
02230240 T  0256
02230300 T  0256
02230400 T  0258
02231000 T  0258
02232000 T  0261
02232010 T  0265
02233000 T  0269
02234000 T  0272
02236000 T  0274
02236010 T  0278
02236020 T  0280
02237000 T  0283
02237010 T  0287
02237012 T  0287
02237015 T  0289
02237020 T  0289
02237021 T  0299
02237030 T  0301
02237040 T  0301
02237050 T  0309
02238000 T  0314
02239000 T  0319
02240000 T  0322
02241000 T  0323
02242000 T  0326
02243000 T  0327
02243010 T  0329
02244000 T  0331
 77 IS  338 LONG, NEXT SEG     6
02245000 T  0492

02245300 T  0492
02245600 T  0492
START OF SEGMENT **********    78

02245900 T  0000
02246200 T  0000
02246500 T  0001
02246800 T  0002
02246900 T  0004
02247000 T  0005
02247100 T  0007
02247200 T  0009
02247400 T  0010
02247700 T  0010
02247800 T  0010
02247900 T  0011
02248000 T  0012
02248300 T  0014
02248600 T  0015
02248900 T  0015
```

```
                    END SCAN;
               REAL STREAM PROCEDURE GETVOID(BUF,VOIDSEQ,FR); VALUE BUF,FR;
PRT(575) = GETVOID
                    BEGIN LABEL L,LC,LD,LE,XIT; LOCAL TA;
                          SI := BUF; DI:=VOIDSEQ; DS:=8LIT" "; DI:=VOIDSEQ;
                    L:    IF SC = " " THEN BEGIN SI←SI+1; GO TO L;  END;
                          TA ← SI;
                          IF SC = """ THEN
                          BEGIN 9(SI←SI+1;
                               IF SC = """ THEN BEGIN SI←SI+1; JUMP OUT TO LC; END
                               ELSE TALLY ← TALLY + 1);
                                TALLY:=TALLY+63; SI:=TA; SI:=SI+1; GO TO LE;
                          END;
                          IF SC < "0" THEN GO TO LD;
                          DS:=8LIT"0";                                        %115-
                          8(SI:=SI+1; DI:=DI-1; TALLY:=TALLY+1;               %115-
                            IF SC LSS "0" THEN JUMP OUT);                     %115-
                     LC:   SI ← TA;
                     LE:   TA:=TALLY; DS:=TA CHR; GETVOID:=SI; GO TO XIT;
                     LD:   GETVOID:=SI;
                          FR(DS:=8LIT"9");
                    XIT:
                    END GETVOID;
               REAL STREAM PROCEDURE SEQNUM(BUF,VLU); VALUE BUF;
PRT(576) = SEQNUM
                    BEGIN LABEL L,LA,LC; LOCAL TA,TB;
                          SI ← BUF;
                    L:    IF SC = " " THEN BEGIN SI←SI+1; GO TO L; END;
                          IF SC = "," THEN BEGIN SI←SI+1; GO TO L; END;
                          TA ← SI;
                          IF SC = "+" THEN
                          BEGIN SI←SI+1;
                          LA:       IF SC = " " THEN BEGIN SI←SI+1; GO TO LA;END;
                                    TB ← SI;
                                    IF SC < "0" THEN BEGIN SI←TA; GO TO LC; END;
                                    DI←TB;TA←DI;
                          END;
                          8(IF SC < "0" THEN JUMP OUT TO LC;
                          TALLY←TALLY+1; SI←SI+1;);
                     LC:  TB ← TALLY; SEQNUM ← SI;
                          SI ← TA; DI ← VLU; DS ← TB OCT;
                    END SEQNUM;
               REAL STREAM PROCEDURE MKABS(S);
PRT(577) = MKABS
                    BEGIN SI ← S; SI←SI+1; MKABS ← SI;
                          DI ← S; 9(DI ← DI + 8); DS ← LIT "[";
                    END MKABS;
               STREAM PROCEDURE MOVEW(P,Q); VALUE Q;
PRT(600) = MOVEW
               BEGIN SI←P; DI ← Q; DS←CHR; END    ;
                    REAL BUF,ID;
STACK(F+2) = BUF
STACK(F+3) = ID
                    BOOLEAN VAL, SAVELISTOG;                                  %511-
STACK(F+4) = VAL
STACK(F+5) = SAVELISTOG
                    LABEL LP,SET,RESET;
```

```
02249200 T  0016
02249500 T  0017
02249800 T  0017
02250100 T  0018
02250400 T  0020
02250500 T  0021
02250700 T  0021
02251000 T  0022
02251300 T  0023
02251600 T  0025
02251900 T  0025
02252200 T  0026
02252500 T  0026
02252600 C  0027
02252800 P  0028
02252850 C  0029
02252900 T  0031
02253100 T  0031
02253400 T  0033
02253700 T  0034
02254000 T  0036
02254300 T  0036
02254600 T  0038
02254900 T  0038
02255200 T  0038
02255500 T  0038
02255800 T  0040
02256100 T  0042
02256400 T  0042
02256700 T  0042
02257000 T  0043
02257300 T  0045
02257600 T  0045
02257900 T  0047
02258200 T  0047
02258500 T  0047
02258800 T  0049
02259100 T  0049
02259400 T  0050
02259700 T  0051
02260000 T  0052
02260300 T  0052
02260600 T  0053
02260900 T  0055
02260925 T  0056
02260950 T  0056
02261200 T  0058
02261500 P  0058
02261800 T  0058
```

```
                    FORMAT WARN(X18,A6," ILLEGAL CONSTRUCT ON DOLLAR CARD XXXX",X39,          02262100 T  0058
                                                                          START OF SEGMENT **********   79
PRT(601) = WARN
                         "WARNING");                                                      02262400 T  0058
                                                                                79 IS   15 LONG, NEXT SEG   78
                    DEFINE GETID = BEGIN ID← "      "; BUF ← SCAN(BUF,ID) END#;          02262700 T  0058
                         SAVELISTOG := LISTOG;                                  %511-    02262775 C  0058
                         MOVEW(CRD[9],BUFL);                                             02262800 T  0059
                         BUF ← MKABS(CRD[0]);                                            02263000 T  0060
                         GETID;                                                          02263300 T  0062
                         IF ID = "VOID  " THEN                                           02263600 T  0065
                            BEGIN BUF ← GETVOID(BUF, VOIDSEQ,0); VOIDTOG ← TRUE          02263900 T  0066
                            END ELSE                                                     02264200 T  0069
                         IF ID = "VOIDT " THEN                                           02264500 T  0071
                            BEGIN BUF ← GETVOID(BUF, VOIDTSEQ,0); VOIDTTOG ← TRUE        02264800 T  0074
                            END ELSE                                                     02265100 T  0078
                    BEGIN                                                                02265400 T  0079
                          TAPETOG,[47:1] ← LASTMODE = 2;                    %517-    02265700 P  0081
                         IF ID = "SET   " OR ID = "+     "  THEN                       02267200 T  0083
               SET;         BEGIN GETID; VAL← TRUE END                                  02267500 T  0085
                         ELSE                                                            02267800 T  0089
                         IF ID = "RESET " OR ID = "-     "  THEN                       02268100 T  0089
               RESET;       BEGIN GETID; VAL ← FALSE END                                02268400 T  0094
                         ELSE                                                            02268700 T  0099
                         BEGIN                                                           02269000 T  0099
                         TSSMESTOG ← VAL; TSSFDITOG ← VAL; CHECKTOG ← VAL;  %501-    02269300 P  0103
                         SINGLETOG←NOT VAL;  HOLTOG←VAL;                    %501-    02269600 P  0108
                         LISTOG←VAL; CODETOG ← DEBUGTOG ← VAL; NEWTPTOG ← VAL;            02269900 T  0112
                         PRTOG←VAL;  DOLIST←VAL;  LIBTAPE←VAL;  SEGPTOG←VAL;  %501-    02270200 P  0116
                         LISTPTOG ← VAL; FREEFTOG ← XREF ← VAL ;                          02270500 T  0123
                         VAL ← TRUE;                                                     02270800 T  0127
                         END;                                                            02271100 T  0128
               LP:       IF ID > 0 THEN                                                  02271400 T  0128
                         BEGIN                                                           02271700 T  0129
                         IF ID = "TRACE " THEN                                           02272000 T  0130
                            BEGIN PRTOG←VAL; LISTOG←VAL; CODETOG←DEBUGTOG←VAL;           02272300 T  0131
                            END ELSE                                                     02272400 T  0136
                         IF ID = "CARD  " THEN                                           02272500 T  0136
                            BEGIN TAPETOG,[47:1]←NOT VAL; LASTMODE←2-REAL(VAL) END ELSE  02272510 P  0138
                         IF ID = "TAPE  " THEN                                           02272550 T  0142
                            BEGIN TAPETOG,[47:1] ← VAL; LASTMODE ← REAL(VAL)+1; END ELSE  02272560 T  0144
                         IF ID = "NOSEQ " THEN SEQTOG ← FALSE ELSE                       02272600 T  0148
                         IF ID = "SET   " OR ID = "+     "  THEN GO TO SET ELSE       02272900 T  0153
                         IF ID = "RESET " OR ID = "-     "  THEN GO TO RESET ELSE     02273200 T  0156
                         IF ID = "ONSITE" AND NOT REMFIXED THEN REMOTETOG ← FALSE ELSE    02273500 T  0159
                         IF ID = "REMOTE" AND NOT REMFIXED THEN REMOTETOG ← VAL ELSE      02273800 T  0164
                         IF ID = "FREEFO" THEN FREEFTOG ← VAL ELSE                       02274100 T  0174
                         IF ID = "SINGLE" OR ID = "SGL   " THEN                          02274400 T  0179
                            BEGIN SINGLETOG ← VAL; LISTOG ← TRUE; END ELSE               02274700 T  0182
                         IF ID = "NEW   " OR ID = "NEWTAP" THEN                          02275000 T  0186
                            BEGIN LIBTAPE←VAL; NEWTPTOG←VAL; NTAPTOG←TRUE;    %501-    02275100 P  0191
                            IF ID = "NEW   " THEN                             %501-    02275120 C  0197
                               BEGIN                                          %501-    02275140 C  0198
                               GETID;                                         %501-    02275160 C  0198
                               IF ID ≠ "TAPE  " THEN                          %501-    02275180 C  0201
                                    GO TO LP;                                 %501-    02275200 C  0202
                               END;                                           %501-    02275220 C  0202
```

```
   END ELSE                                                          %501-    02275240 C  0202
IF ID = "LIST  " THEN LISTOG ← VAL ELSE                                      02275300 T  0202
IF ID = "SEQXEQ" AND NOT SEGSWFIXED THEN SEGSW ← VAL ELSE                    02275600 T  0211
IF ID = "PRT   " THEN PRTOG ← VAL ELSE                                       02275900 T  0217
IF ID = "DEBUGN" THEN                                                        02276200 T  0222
   BEGIN LISTOG←VAL; CODETOG←VAL; PRTOG ← VAL END ELSE                       02276500 T  0224
IF ID = "TIME  " THEN TIMETOG ← VAL ELSE                                     02276800 T  0229
IF ID = "ERRMES" THEN TSSMESTOG ← VAL ELSE                                   02277100 T  0234
IF ID = "TSSEDI" THEN TSSEDITOG ← VAL ELSE                                   02277400 T  0239
IF ID = "LISTLI" THEN LISTLIBTOG ← VAL ELSE                                  02277700 T  0244
IF(ID = "SEGMEN" OR ID = "SEG   ") AND VAL THEN                              02278000 T  0249
   BEGIN ADR←ADR+1; SEGOVF;END ELSE                                          02278300 T  0253
IF ID = "PAGE  " AND VAL THEN WRITE(LINE[PAGE]) ELSE                         02278600 T  0255
IF ID = "VOID  " THEN                                                        02278900 T  0263
   BEGIN VOIDTOG ← VAL;                                                      02279200 T  0266
         IF VAL THEN BUF ← GETVOID(BUF,VOIDSEQ,1);                           02279300 T  0269
   END ELSE                                                                  02279500 T  0272
IF ID = "VOIDT " THEN                                                        02279800 T  0272
  BEGIN VOIDTTOG ← VAL;                                                      02280100 T  0274
         IF VAL THEN BUF ← GETVOID(BUF,VOIDTSEQ,1);                          02280300 T  0277
  END ELSE                                                                   02280400 T  0280
IF ID = "LIMIT " THEN                                                        02280700 T  0280
   BEGIN LIMIT := IF VAL THEN 0 ELSE @60;                                    02281000 T  0282
         BUF := SEQNUM(BUF,LIMIT);                                           02281300 T  0285
         IF LIMIT LEQ ERRORCT THEN GO TO POSTWRAPUP;                         02281600 T  0288
   END ELSE                                                                  02282500 T  0291
IF ID = "XREF  " THEN                                                        02282800 T  0291
   BEGIN                                                                     02282850 T  0294
         PXREF ← TRUE; XREF ← VAL;                                           02282900 T  0295
   END ELSE                                                                  02282950 T  0297
                        IF ID = "SEQ   " THEN                                02283100 T  0297
                        BEGIN SEQTOG ← VAL;                                  02283200 T  0300
                              IF VAL THEN                                    02283300 T  0303
                           BEGIN SEQBASE ← SEQINCR ← 0;                      02283400 T  0303
                                 BUF ← SEQNUM(BUF,SEQBASE);                  02283700 T  0305
                                 BUF ← SEQNUM(BUF,SEQINCR);                  02284000 T  0307
                                 IF SEQINCR ≤ 0 THEN SEQINCR ← 1000;         02284600 T  0310
                        END END ELSE                                         02284900 T  0312
                        IF ID = "LISTDO" THEN DOLIST ← VAL ELSE              02285200 T  0312
                        IF ID = "HOL   " THEN HOLTOG ← VAL ELSE              02285500 T  0317
                        IF ID = "CHECK " THEN CHECKTOG ← VAL ELSE            02285800 T  0322
                        IF ID = "NEWPAG" THEN SEGPTOG ← VAL ELSE     %501-   02286100 P  0327
                        IF ID = "LISTP " THEN LISTPTOG ← VAL ELSE            02286400 T  0332
                        BEGIN IF FIRSTCALL THEN DATIME;                      02286700 T  0337
                              IF UNPRINTED THEN BEGIN PRINTCARD; UNPRINTED←FALSE END;  02287000 P  0340
                              IF SINGLETOG THEN WRITE(LINE,WARN,ID)          02287300 T  0344
PRT(602) = *LIST, LABEL, OR SEGMENT DESCRIPTOR*
                              ELSE WRITE(RITE,WARN,ID);                      02287600 T  0350
PRT(603) = *LIST, LABEL, OR SEGMENT DESCRIPTOR*
                        END                                                  02287900 T  0361
                        ;                                                    02288200 T  0361
                        GETID;                                               02288500 T  0361
                        GO TO LP;                                            02288800 T  0364
                        END;                                                 02289100 T  0366
                     END;                                                    02289400 T  0366
                        IF DOLIST OR LISTOG OR SAVELISTOG THEN       %511-   02289450 C  0366
                           IF UNPRINTED THEN PRINTCARD;              %517-   02289500 P  0368
```

```
                    UNPRINTED ← TRUE;                                                    02289600 T  0370
                END DOLOPT;                                                              02289700 T  0372
                STREAM PROCEDURE NEWSEQ(A,B); VALUE B;                                78 IS  375 LONG, NEXT SEG    6
                                                                                         02317000 T  0492
PRT(604) = NEWSEQ                                                                        02318000 T  0492
                    BEGIN                                                                02319000 T  0492
                        SI ← LOC B; DI ← A; DS ← 8 DEC;                                  02320000 T  0492
                    END NEWSEQ;                                                          02321000 T  0493
                INTEGER STREAM PROCEDURE SEQCHK(T,C);
PRT(605) = SEQCHK                                                                        02322000 T  0493
                    BEGIN                                                                02323000 T  0493
                        SI ← T; DI ← C;                                                  02324000 T  0493
                        IF 8 SC < DC THEN TALLY ← 4 ELSE                                 02325000 T  0494
                    BEGIN                                                                02326000 T  0494
                        SI ← SI - 8; DI ← DI - 8;                                        02327000 T  0495
                        IF 8 SC =DC THEN TALLY ← 2                                       02328000 T  0495
                                    ELSE TALLY ← 3;                                      02329000 T  0496
                    END;                                                                 02330000 T  0496
                        SEQCHK ← TALLY;                                                  02331000 T  0496
                    END SEQCHK;                                                          02332000 T  0497
                BOOLEAN PROCEDURE READACARD;                                             02333000 T  0497
                BEGIN                                                                    02333050 T  0497
                DEFINE FLAGI(FLAGI1) = BEGIN FLAG(FLAGI1); GO TO E4A;END #;           START OF SEGMENT **********   80
                REAL STREAM PROCEDURE SCANINC(BUF,ID,RESULT,N,M);                        02333100 T  0000
PRT(606) = SCANINC                                                                       02333120 T  0000
                    VALUE BUF,M,N;                                                       02333140 T  0000
                    BEGIN                                                                02333160 T  0000
                        LOCAL TA;                                                        02333180 T  0000
                        LABEL LP,LQ,XIT;                                                 02333200 T  0000
                        DI := RESULT; DI := DI +7;                                       02333210 T  0000
                        SI := BUF;                                                       02333220 T  0000
                     LP: IF SC = " " THEN BEGIN SI := SI + 1; GO TO LP; END;             02333240 P  0002
                        IF SC = ALPHA THEN ELSE BEGIN DS:=LIT "1"; DI:=ID; DS:=2LIT"0";  02333260 T  0004
                            DS := CHR; DS := 5LIT" ";GO TO XIT; END;                     02333270 C  0006
                        IF SC LSS "0" THEN DS:=LIT "2" ELSE DS:=LIT "3";          %400-  02333280 P  0007
                        N (DI:=ID; DS:=8 LIT "0       "; DI:=DI-7;                %400-  02333300 T  0010
                           7(IF SC=ALPHA THEN DS:=CHR ELSE JUMP OUT 2 TO XIT);            02333320 T  0012
                         JUMP OUT TO XIT);                                               02333360 P  0013
                        M ( 8 ( IF SC LSS "0" THEN JUMP OUT 2 TO LQ;              %400-  02333370 C  0016
                                IF SC GTR "9" THEN JUMP OUT 2 TO LQ;              %400-  02333380 P  0017
                              SI:=SI+1; TALLY:=TALLY+1));                         %400-  02333400 T  0018
                     LQ: TA := TALLY;                                                    02333420 T  0019
                        SI := SI - TA;                                                   02333440 T  0019
                        DI := ID;                                                        02333460 T  0020
                        DS := TA OCT;                                                    02333500 T  0020
                    XIT:  SCANINC := SI;                                                 02333520 T  0021
                    END SCANINC;                                                         02333540 T  0022
                REAL STREAM PROCEDURE MKABS(S);
PRT(607) = MKABS                                                                         02333550 T  0022
                    BEGIN SI := S; SI := SI + 1;  MKABS := SI;  END;                     02334000 T  0024
                STREAM PROCEDURE MOVE(P, Q); VALUE Q;
PRT(610) = MOVE                                                                          02335000 T  0024
                BEGIN SI ← P; DI ← Q; DS ← CHR;                                          02336000 T  0025
                   DI ← P; DS ← LIT "]";                                                 02337000 T  0026
                END MOVE;                                                                02337100 T  0026
                STREAM PROCEDURE MOVEC(C,B);    VALUE B;
```

```
PRT(611) = MOVEC
                BEGIN SI←B; DI←C; DS←CHR; END;                                                  02337200 T  0026
                BOOLEAN STREAM PROCEDURE GETCOL1(BUF);                                          02338000 T  0028
PRT(612) = GETCOL1
                     BEGIN
                          SI ← BUF; IF SC = "$" THEN TALLY ← 1;                                 02339000 T  0028
                          GETCOL1 ← TALLY;                                                      02340000 T  0028
                     END;                                                                       02341000 T  0029
                STREAM PROCEDURE TSSEDITS(C,P);  BEGIN SI←C;  DI←P;  DS←10WDS;  SI←C ;          02342000 T  0029
PRT(613) = TSSEDITS                                                                             02342010 T  0030
                IF SC="C" THEN BEGIN DI←P;  DI←DI+1;  DS←LIT"=" END ELSE                        02342020 T  0032
                IF SC≠"$" THEN BEGIN SI←SI+5;  IF SC≠" " THEN IF SC≠"0" THEN BEGIN DI←P;        02342030 T  0034
                DS←6LIT"=     " END END END OF TSSEDITS ;                                       02342040 T  0037
                STREAM PROCEDURE MOVEW(F,T,B,R); VALUE B, R;                                    02343000 T  0039
PRT(614) = MOVEW
                     BEGIN                                                                      02344000 T  0039
                          LABEL XIT;                                                            02345000 T  0039
                          SI ← F; DI ← T;                                                       02346000 T  0039
                          B(                                                                    02347000 T  0039
                          2(40( IF SC = ALPHA THEN DS ← CHR ELSE                                02348000 T  0040
                               IF SC = " " THEN DS ← CHR ELSE                                   02349000 T  0042
                               IF SC = "%" THEN BEGIN DS ← LIT "("; SI ← SI+1 END ELSE          02350000 T  0043
                               IF SC = "[" THEN BEGIN DS ← LIT ")"; SI ← SI+1 END ELSE          02351000 T  0045
                               IF SC = "#" THEN BEGIN DS ← LIT "="; SI ← SI+1 END ELSE          02352000 T  0047
                               IF SC = "&" THEN BEGIN DS ← LIT "+"; SI ← SI+1 END ELSE          02353000 T  0049
                               IF SC = "@" THEN BEGIN DS ← LIT """; SI ← SI+1 END ELSE          02354000 T  0051
                               IF SC = ":" THEN BEGIN DS ← LIT """; SI ← SI+1 END ELSE          02354100 T  0053
                               IF SC = "<" THEN BEGIN DS ← LIT "+"; SI ← SI+1 END ELSE          02355000 T  0055
                               IF SC = ">" THEN BEGIN DS ← LIT "="; SI ← SI+1 END ELSE          02356000 T  0057
                          DS ← CHR ));  JUMP OUT TO XIT);                                       02357000 T  0059
PRT(615) = *LIST, LABEL, OR SEGMENT DESCRIPTOR*
PRT(616) = *LIST, LABEL, OR SEGMENT DESCRIPTOR*
PRT(617) = *LIST, LABEL, OR SEGMENT DESCRIPTOR*
                          DS ← 10 WDS;                                                          02358000 T  0062
                     XIT;                                                                       02359000 T  0062
                          SI←LOC R;  SI←SI+7;  DI←DI+1 ;                                        02360000 T  0062
                          DS←CHR ;                                                              02361000 T  0063
                     END MOVEW;                                                                 02362000 T  0064
                ALPHA STREAM PROCEDURE DCMOVEW(F,T,B,FIL,R); VALUE B,FIL,R;                     02362010 T  0064
PRT(620) = DCMOVEW
                     BEGIN LOCAL C;  LABEL L1,L2,L3,L4 ;                                        02362020 T  0064
                     SI←F;  DI←T;  2(SI←SI+36;  DS←36LIT" ");  C←SI;  DS←8CHR ;                 02362030 T  0065
                     SI←LOC   R; SI←SI+6; DS←2CHR; DI←C; DS←8LIT"]";TALLY←33;SI←F;               02362033 T  0071
                     IF SC = " " THEN     %%% IT MIGHT BE A FILES CARD.                         02362034 T  0074
                          BEGIN SI←SI+1; IF SC="F" THEN                                         02362035 T  0074
                          BEGIN     DI←LOC FIL; DI←DI+2; IF 5SC=DC THEN                         02362036 T  0076
                          BEGIN DI←F; SI←LOC FIL; SI←SI+2; DS←6CHR ; GO TO L1; END              02362037 T  0077
                          ELSE SI←F END ELSE SI←F  END                                          02362038 T  0079
                     ELSE IF SC ="F" THEN BEGIN DI←LOC FIL;DI←DI+2;IF 6SC=DC THEN GO L1         02362040 T  0080
                               ELSE SI←F;END                                                    02362041 T  0082
                     ELSE IF SC="=" THEN                                                        02362042 T  0083
                          BEGIN %%% IT IS A CONTINUATION CARD.                                  02362044 T  0084
                          SI←SI+1;  DI←T;  DS←6LIT"     *" ;                                   02362050 T  0084
                          5(IF SC=" " THEN SI←SI+1 ELSE JUMP OUT);  GO TO L2 ;                  02362060 T  0086
                          END                                                                   02362063 T  0088
                     ELSE IF SC="C" THEN                                                        02362066 T  0088
                          BEGIN %%% IT MIGHT BE A COMMENT CARD.                                 02362070 T  0089
```

```
                    SI←SI+1;  IF SC="=" THEN GO TO L1 ELSE SI←F ;                                  02362080 T  0089
                    END ;                                                                          02362090 T  0091
               IF SC≠"$" THEN %%% IT IS NOT A COMMENT CARD, NOR IS IT A $ CARD,                     02362100 T  0091
                    BEGIN     %%% NOR A CONTINUATION CARD, NOR A FILES CARD,                       02362110 T  0091
                    2(33(IF SC=" " THEN SI←SI+1 ELSE JUMP OUT 2 TO L3)); L3:  DI←T;                02362120 T  0091
                    5(IF SC=" " THEN SI←SI+1 ELSE IF SC≥"0" THEN IF SC≤"9"                         02362130 T  0095
                      THEN DS←CHR ELSE JUMP OUT ELSE JUMP OUT) ;                                   02362140 T  0097
                    DI←T;  DI←DI+6;  IF SC=" " THEN SI←SI+1 ;                                      02362150 T  0100
                    END                                                                            02362160 T  0101
               ELSE                                                                                02362170 T  0101
                    BEGIN                                                                          02362180 T  0101
                    L1:  SI←F;  DI←T;  TALLY←36 ;                                                  02362190 T  0101
PRT(621) = *LIST, LABEL, OR SEGMENT DESCRIPTOR*
                    END ;                                                                          02362200 T  0102
               L2:  C←TALLY ;                                                                      02362210 T  0102
               B(2(C((IF SC>">" THEN DS←CHR ELSE IF SC<"[" THEN DS←CHR ELSE                        02362220 T  0103
                     IF SC="%" THEN BEGIN DS←LIT"("; SI←SI+1 END ELSE                              02362230 T  0107
                     IF SC="#" THEN BEGIN DS←LIT"="; SI←SI+1 END ELSE                              02362235 T  0109
                     IF SC="&" THEN BEGIN DS←LIT"+"; SI←SI+1 END ELSE                              02362240 T  0111
                     IF SC="@" THEN BEGIN DS←LIT""""; SI←SI+1 END ELSE                             02362245 T  0113
                     IF SC=":" THEN BEGIN DS←LIT""""; SI←SI+1 END ELSE                             02362250 T  0115
                     IF SC="<" THEN BEGIN DS←LIT"+"; SI←SI+1 END ELSE                              02362255 T  0117
                     IF SC=">" THEN BEGIN DS←LIT"="; SI←SI+1 END ELSE                              02362260 T  0119
                     IF SC="[" THEN BEGIN DS←LIT")"; SI←SI+1 END ELSE                              02362265 T  0121
                     IF SC="]" THEN JUMP OUT 3 TO L4 ELSE DS←CHR)); JUMP OUT TO L4);               02362270 T  0123
PRT(622) = *LIST, LABEL, OR SEGMENT DESCRIPTOR*
PRT(623) = *LIST, LABEL, OR SEGMENT DESCRIPTOR*
PRT(624) = *LIST, LABEL, OR SEGMENT DESCRIPTOR*
PRT(625) = *LIST, LABEL, OR SEGMENT DESCRIPTOR*
PRT(626) = *LIST, LABEL, OR SEGMENT DESCRIPTOR*
                    2(C((IF SC="]" THEN JUMP OUT 2 TO L4 ELSE DS←CHR)) ;                           02362275 T  0128
                    L4:  DI←LOC DCMOVEW;  DS←8LIT"00      ";  DI←DI+6 ;                            02362280 T  0131
                    6(IF SC="]" THEN JUMP OUT ELSE DS←CHR) ;                                       02362285 T  0133
                    END OF DCMOVEW ;                                                               02362290 T  0136
               STREAM PROCEDURE SEQERR(BUFF, NEW, OLD);                                            02362300 T  0137
PRT(627) = SEQERR
               BEGIN                                                                               02362400 T  0137
                    DI ← BUFF; DS ← 18 LIT "SEQUENCE ERROR    ";                                   02362450 T  0137
                    SI ← NEW; DS ← LIT"""";                                                        02362500 T  0139
                    DS ← 8 CHR; DS ← LIT """";                                                     02362550 T  0140
                    DS ← 3 LIT " < ";                                                              02362600 T  0141
                    SI ← OLD; DS ← LIT """";                                                       02362650 T  0142
                    DS ← 8 CHR; DS ← LIT """";                                                     02362700 T  0142
                    DS ← 59 LIT " ";                                                               02362750 T  0143
                    DS ← 8 LIT "X"; DS ← 4 LIT " ";                                                02362800 T  0151
               END SEQERR;                                                                         02362850 T  0153
               STREAM PROCEDURE DCMOVE(E,C,A,N); VALUE A,N;                                        02362855 T  0153
PRT(630) = DCMOVE
               BEGIN SI←C; DI←E; DS←10WDS; DS←4CHR; SI←LOC A; DS←4DEC;                             02362860 T  0153
                  N(DS←8LIT"   PATCH"); END;                                                       02362865 T  0155
               READ BUF,ID;                                                                        02362900 T  0158
STACK(F+3) = BUF
STACK(F+4) = ID
               BOOLEAN NOWRI;  LABEL ENDPB, E4B;                                                   02362902 T  0158
STACK(F+5) = NOWRI
               LABEL LIBADD, ENDPA, E4A, E4, STRTA;                                                02362950 T  0158
               LABEL E1,E2,E3,STRT,ENDP,XIT;                                                       02363000 T  0158
```

```
      UNPRINTED←TRUE ;                                                                  02363100 T  0158
      GO TO STRT;                                                                       02364000 T  0160
LIBADD:                                                                                 02364100 T  0161
      XTA := BLANKS;                                                                    02364130 T  0162
      IF INSERTDEPTH = -1 THEN SAVECARD := NEXTCARD;                                    02364160 T  0162
      MOVE (CRD[9],BUFL);                                                               02364240 T  0165
                                                                                        02364280 T  0166
      IF (INSERTDEPTH := INSERTDEPTH + 1) GTR INSERTMAX THEN                            02364300 T  0166
           FLAGI(158);                                                                  02364320 T  0168
      NEWTPTOG := FALSE;                                                                02364330 T  0169
      INSERTINX := -1;                                                                  02364340 T  0171
      INSERTCOP ← 0;                                                                    02364350 T  0173
      BLANKIT(SSNM[5],1,0);                                                             02364360 T  0175
      BUF := SCANINC(BUF,INSERTMID,RESULT,1,0);                                         02364380 T  0177
      IF RESULT = 1 AND INSERTMID = "+     "THEN                                        02364381 T  0181
      BEGIN BUF←SCANINC(BUF,ID,RESULT,1,0);                                             02364382 T  0184
            IF ID = "COPY    " THEN INSERTCOP := 1                                      02364383 T  0187
            ELSE FLAGI(155);                                                            02364384 T  0190
            BUF←SCANINC(BUF,INSERTMID,RESULT,1,0);                                      02364385 T  0195
      END;                                                                              02364386 T  0199
      IF RESULT NEQ 2 THEN FLAGI (155);                                                 02364400 T  0199
           BUF ← SCANINC(BUF,ID,RESULT,0,1);                                    %107-   02364420 P  0201
      IF RESULT = 1  THEN IF ID = "/      " THEN                                        02364440 T  0204
      BEGIN BUF ← SCANINC(BUF,INSERTFID,RESULT,1,0);                                    02364460 T  0206
            IF RESULT NEQ 2 THEN FLAGI(155);                                            02364480 T  0211
            BUF ← SCANINC(BUF,ID,RESULT,0,1);                                           02364500 T  0215
       END ELSE INSERTFID := TIME(-1) ELSE INSERTFID := TIME(-1);                       02364520 T  0218
      IF RESULT = 3 THEN                                                                02364540 T  0224
       BEGIN NEWSEQ(SSNM[5],ID);                                                        02364560 T  0225
            BUF ← SCANINC(BUF,ID,RESULT,0,1);                                           02364580 T  0227
             IF RESULT NEQ   1 THEN FLAGI(156);                                         02364600 T  0230
             IF ID = "]     " THEN NEWSEQ (INSERTSEQ,99999999)             %400-       02364605 C  0232
             ELSE BEGIN                                                     %400-       02364610 C  0235
            BUF ← SCANINC(BUF,ID,RESULT,0,1);                                           02364620 T  0239
             IF RESULT NEQ   3 THEN FLAGI(157);                                         02364640 T  0242
             NEWSEQ (INSERTSEQ,ID);                                                     02364660 T  0244
             END                                                            %400-       02364670 C  0246
       END ELSE IF ID = "]     " THEN NEWSEQ(INSERTSEQ,99999999)                        02364680 T  0246
           ELSE FLAGI(157);                                                             02364700 T  0249
      IF INSERTDEPTH > 0 THEN CLOSE (LF,RELEASE);                                       02364720 T  0254
      FILL LF WITH      INSERTMID,INSERTFID;                                            02364740 T  0257
      DO BEGIN                                                                          02364760 T  0263
           READ (LF[INSERTINX ← INSERTINX+1],10,DB[*])[E4];                             02364780 T  0263
PRT(631) = E4                                                                           02364800 T  0272
           END UNTIL SEQCHK(SSNM[5],DB[9]) NEQ 3;                                       02364810 T  0275
      NEWTPTOG ← BOOLEAN(INSERTCOP);                                                    02364811 T  0278
      IF NOT NEWTPTOG THEN                                                              02364815 T  0279
      BEGIN                                                                             02364820 T  0280
      IF SEQTOG THEN                                                                    02364830 T  0280
      BEGIN NEWSEQ(CRD[9],SEQBASE); MOVE(CRD[9],BUFL);                                  02364840 T  0283
            SEQBASE←SEQBASE+SEQINCR;                                                    02364850 T  0285
      END;     MOVEC(CRD[9],BUFL);                                                      02364860 T  0286
      IF LIBTAPE AND(INSERTDEPTH = 0 ) THEN WRITE(NEWTAPE,10,CRD[*]);                   02364885 P  0291
      END;                                                                  %511-       02364890 P  0292
      IF LISTOG OR DOLIST THEN PRINTCARD;                                   %511-       02364900 T  0295
      NEXTCARD:=7;                                                                      02364910 T  0296
      GO TO STRT;
```

```
                    E1:  IF NEXTCARD=1 THEN IF TAPETOG THEN                                  02365000 T  0296
                            BEGIN                                                            02366000 T  0298
                            NEXTCARD←5;  READ(TP,10,TB[*])[E2];  GO TO STRT ;                02367000 T  0299
PRT(632) = E2
                            END                                                              02368000 T  0305
                         ELSE                                                                02369000 T  0305
                            BEGIN                                                            02370000 T  0305
                             NEXTCARD:=6;                                                    02371000 T  0306
                         IF GETCOL1(CRD[0]) THEN BEGIN BUF := MKABS(CRD[0]);                 02371002 T  0306
                            BUF←SCANINC(BUF,ID,RESULT,1,0); IF ID NEQ INCLUDE                02371004 T  0310
                            THEN BLANKIT(CRD,9,1); END;                                      02371006 T  0314
                            GO TO ENDP ;                                                     02371010 T  0316
                            END ;                                                            02371020 T  0317
                         NEXTCARD ← 5; GO TO ENDP;                                           02372000 T  0317
                    E2:  IF NEXTCARD=5 THEN                                                  02372010 T  0318
                            BEGIN                                                            02372020 T  0319
                             NEXTCARD:=6;                                                    02372030 T  0320
                         IF GETCOL1(CRD[0]) THEN BEGIN BUF := MKABS(CRD[0]);                 02372040 T  0321
                            BUF←SCANINC(BUF,ID,RESULT,1,0); IF ID NEQ INCLUDE                02372050 T  0325
                            THEN BLANKIT(CRD,9,1); END;                                      02372060 T  0328
                            END                                                              02373000 T  0330
                         ELSE IF NEXTCARD≠2 THEN NEXTCARD←1 ELSE                             02373010 T  0330
                            BEGIN                                                            02373020 T  0333
                            NEXTCARD←1;  TAPETOG←BOOLEAN(2);  READ(CR,10,CB[*])[E1] ;        02373030 T  0333
PRT(633) = E1
                            END ;                                                            02373040 T  0340
                    IF VOIDTTOG THEN IF SEQCHK(CRD[9],VOIDTSEQ) < 4          %114-           02373045 C  0340
                          THEN VOIDTTOG:=FALSE                               %114-           02373100 C  0343
                           ELSE BEGIN VOIDTTOG:=FALSE; GO TO STRT; END;      %114-           02373200 C  0345
                         GO TO ENDP ;                                                        02374000 T  0349

                    E4B: NOWRI ←TRUE;                                                        02374050 T  0349
                         IF SEQTOG THEN NEWSEQ(CRD[9],SEQBASE);                              02374053 T  0350
                         IF NEWTPTOG THEN IF TSSEDITOG THEN                                  02374055 T  0353
                         BEGIN TSSEDITS(CRD,PRINTBUFF); WRITE(NEWTAPE,10,PRINTBUFF[*]);      02374060 T  0355
                         END ELSE WRITE(NEWTAPE,10,CRD[*]);                                  02374065 T  0361
                    E4:  CLOSE (LF,RELEASE);                                                 02374100 T  0366
                         NEWTPTOG:= SAVETOG;                                                 02374110 T  0368
                         IF (INSERTDEPTH := INSERTDEPTH - 1) = -1 THEN                       02374150 T  0370
                              BEGIN NEXTCARD ← SAVECARD; GO TO ENDPA; END                    02374200 T  0372
                              ELSE NEWTPTOG ← BOOLEAN(INSERTCOP);                            02374300 T  0374
                         FILL LF WITH INSERTMID,INSERTFID;                                   02374450 T  0377
                         READ(LF[INSERTINX := INSERTINX + 1],10,DB[*])[E4];                  02374470 T  0383
                         IF SEQCHK(INSERTSEQ,DB[9]) = 4 THEN GO TO E4;                       02374500 T  0393
                         GO TO ENDP;                                                         02374520 T  0396
                    E4A:                                                                     02374560 T  0397
                         NEWTPTOG←SAVETOG;                                                   02374580 T  0398
                         IF (INSERTDEPTH := INSERTDEPTH-1) = -1 THEN NEXTCARD:=SAVECARD      02374600 T  0399
                         ELSE NEWTPTOG ← BOOLEAN(INSERTCOP);                                 02374680 T  0402
                    STRT:                                                                    02375000 T  0406
                    STRTA:                                                                   02375600 T  0407
                         IF NEXTCARD=6 THEN GO XIT ;                                         02376000 T  0407
                         CARDCOUNT←CARDCOUNT+1 ;                                             02377000 T  0408
                         IF NEXTCARD = 1 THEN                                                02378000 T  0409
                    BEGIN     % CARD ONLY                                                    02379000 T  0410
                         IF(NOT DCINPUT OR TSSEDITOG)AND NOT FREEFTOG                        02380000 T  0410
                         THEN MOVEW(CB,CRD,HOLTOG,IF DCINPUT THEN "D" ELSE "R")              02380100 T  0412
                         ELSE IF SSNM[4]←DCMOVEW(CB,CRD,HOLTOG,"FILE  ",IF FREEFTOG          02380125 T  0419
```

```
                THEN " R" ELSE " D") ≠ "      " THEN                                  02380150 T  0425
            BEGIN XTA←SSNM[4] ;                                                       02380175 T  0428
            MOVESEQ(SSNM[4],LASTSEQ);  MOVESEQ(LASTSEQ,CRD[9]) ;                      02380200 T  0430
            IF LISTOG THEN                                                            02380210 T  0433
                 BEGIN IF FIRSTCALL THEN DATIME ;                                     02380220 T  0434
                 DCMOVE(PRINTBUFF,CRD,(ADR+1),[36:10],0);                             02380230 T  0436
                 WRITAROW(11,PRINTBUFF);  UNPRINTED←FALSE ;                           02380240 T  0439
                 END ;                                                                02380250 T  0442
            FLOG(149);  MOVESEQ(LASTSEQ,SSNM[4]) ;                                    02380300 T  0442
            END ;                                                                     02380400 T  0444
      IF GETCOL1(CRD[0]) THEN                                                         02381000 T  0444
BEGIN                                                                                 02382000 T  0446
      BUF := MKABS(CRD[0]);                                                           02382200 T  0446
      BUF := SCANINC(BUF,ID,RESULT,1,0);                                              02382300 T  0448
      IF ID NEQ INCLUDE THEN                                                          02382400 T  0451
BEGIN                                                                                 02382500 T  0452
      DOLOPT;                                                                         02383000 T  0453
      IF REAL(TAPETOG)=3 THEN READ(TP) ;                                              02383010 T  0453
      READ(CR,10,CB[*])[E1];                                                          02384000 T  0458
      IF NOT TAPETOG THEN GO TO STRT;                                                 02385000 T  0464
      READ(TP,10,TB[*])[E2];                                                          02386000 T  0464
      NEXTCARD ← SEQCHK(TB[9], CB[9]);                                                02387000 T  0470
      GO TO STRT;                                                                     02388000 T  0472
END;                                                                                  02389000 T  0477
END;                                                                                  02389100 T  0477
      IF LISTPTOG THEN                                                                02389200 T  0477
      BEGIN IF FIRSTCALL THEN DATIME;                                                 02389300 T  0477
            IF SEQTOG THEN NEWSEQ(CRD[9],SEQBASE);                                    02389400 T  0480
            DCMOVE(PRINTBUFF,CRD,(ADR+1),[36:10],1);                                  02389500 T  0482
            WRITAROW (12,PRINTBUFF); UNPRINTED := FALSE;                              02389600 T  0485
      END;                                                                            02389700 T  0488
      READ(CR,10,CB[*])[E1];                                                          02390000 T  0488
      GO TO ENDP;                                                                     02391000 T  0494
END;                                                                                  02392000 T  0494
      IF NEXTCARD = 5 THEN                                                            02393000 T  0494
BEGIN                                                                                 02394000 T  0495
      MOVEW(TB,CRD,HOLTOG,"T") ;                                                      02395000 T  0495
       READ(TP, 10, TB[*])[E2];                                                       02396000 T  0499
      IF VOIDTTOG THEN IF SEQCHK(CRD[9],VOIDTSEQ) < 4 THEN                            02396100 T  0504
         VOIDTTOG ← FALSE ELSE GO TO STRT;                                            02396200 T  0507
      GO TO ENDP;                                                                     02397000 T  0510
END;                                                                                  02398000 T  0510
      IF NEXTCARD ≤ 3 THEN                                                            02399000 T  0510
BEGIN     % CARD OVER TAPE                                                            02400000 T  0511
      IF(NOT DCINPUT OR TSSEDITOG)AND NOT FREEFTOG                                    02401000 T  0511
      THEN MOVEW(CB,CRD,HOLTOG,IF DCINPUT THEN "D" ELSE "R")                          02401100 T  0513
      ELSE IF SSNM[4]←DCMOVEW(CB,CRD,HOLTOG,"FILE  ",IF FREEFTOG                      02401125 T  0520
            THEN " R" ELSE " D") ≠ "      " THEN                                      02401150 T  0526
            BEGIN XTA←SSNM[4] ;                                                       02401175 T  0529
            MOVESEQ(SSNM[4],LASTSEQ);  MOVESEQ(LASTSEQ,CRD[9]) ;                      02401200 T  0531
            IF LISTOG THEN                                                            02401210 T  0534
                 BEGIN IF FIRSTCALL THEN DATIME ;                                     02401220 T  0535
                 DCMOVE(PRINTBUFF,CRD,(ADR+1),[36:10],0);                             02401230 T  0537
                 WRITAROW(11,PRINTBUFF);  UNPRINTED←FALSE ;                           02401240 T  0540
                 END ;                                                                02401250 T  0543
            FLOG(149);  MOVESEQ(LASTSEQ,SSNM[4]) ;                                    02401300 T  0543
            END ;                                                                     02401400 T  0545
```

```
                    IF NEXTCARD =2 THEN READ(TP,10,TB[*])[E2];                                  02402000 T  0545
                    IF GETCOL1(CRD) THEN                                                        02403000 T  0552
               BEGIN                                                                            02404000 T  0553
                    BUF := MKABS(CRD[0]);                                                       02404200 T  0554
                         BUF := SCANINC(BUF,ID,RESULT,1,0);                                     02404300 T  0556
                    IF ID NEQ INCLUDE THEN                                                      02404400 T  0559
               BEGIN                                                                            02404500 T  0560
                    DOLOPT;                                                                     02405000 T  0560
                    READ(CR,10,CB[*])[E3];                                                      02406000 T  0561
PRT(634) = E3
                    NEXTCARD ← SEQCHK(TB[9], CB[9]);                                            02407000 T  0566
                    GO TO STRT;                                                                 02408000 T  0568
                    E3: NEXTCARD←5;  GO TO STRT;                                                02408500 T  0574
               END;                                                                             02409000 T  0575
               END;                                                                             02409100 T  0575
                    IF LISTPTOG THEN                                                            02409200 T  0575
                    BEGIN IF FIRSTCALL THEN DATIME;                                             02409300 T  0576
                          IF SEQTOG THEN NEWSEQ(CRD[9],SEQBASE);                                02409400 T  0578
                          DCMOVE(PRINTBUFF,CRD,(ADR+1).[36:10],1);                              02409500 T  0580
                          WRITAROW (12,PRINTBUFF); UNPRINTED := FALSE;                          02409600 T  0583
                    END;                                                                        02409700 T  0587
                    READ(CR,10,CB[*])[E1];                                                      02410000 T  0587
                    NEXTCARD ← SEQCHK(TB[9], CB[9]);                                            02411000 T  0592
                    GO TO ENDP;                                                                 02412000 T  0594
               END;                                                                             02413000 T  0595
                         % TAPE BEFORE CARD                                                     02414000 T  0595
                    IF NEXTCARD = 7 THEN                                                        02414100 T  0595
               BEGIN                                                                            02414150 T  0596
                    IF(NOT DCINPUT OR TSSEDITOG)AND NOT FREEFTOG                                02414175 T  0596
                    THEN MOVEW(DB,CRD,HOLTOG,"L") ELSE IF XTA←DCMOVEW(DB,CRD,                   02414200 T  0598
                         HOLTOG,"FILE  ",IF FREEFTOG THEN " R" ELSE " D")                       02414225 T  0607
                           ≠ "       " THEN FLOG(149);                                          02414250 T  0610
                    IF GETCOL1(CRD[0]) THEN                                                     02414260 T  0613
               BEGIN                                                                            02414270 T  0614
                    BUF := MKABS(CRD[0]);                                                       02414280 T  0615
                         BUF := SCANINC(BUF,ID,RESULT,1,0);                                     02414290 T  0617
                    IF ID = INCLUDE THEN GO TO LIBADD;                                          02414300 T  0620
               END;                                                                             02414310 T  0621
                    READ(LF[INSERTINX←INSERTINX+1],10,DB[*])[E4B];                              02414370 T  0621
PRT(635) = E4B
                    IF SEQCHK(INSERTSEQ,DB[9]) = 4 THEN GO TO E4B;                              02414380 T  0631
                    IF GETCOL1(CRD[0]) THEN BEGIN DOLOPT; GO TO STRT; END;                      02414382 T  0635
                    GO TO ENDP;                                                                 02414390 T  0642
               END;                                                                             02414400 T  0642
                    MOVEW(TB,CRD,HOLTOG,"T") ;                                                  02415000 T  0642
                    READ(TP,10,TB[*])[E2];                                                      02416000 T  0645
                    IF VOIDTTOG THEN IF SEQCHK(CRD[9],VOIDTSEQ) < 4 THEN                        02416100 T  0651
                      VOIDTTOG:=FALSE ELSE BEGIN NEXTCARD:=SEQCHK(TB[9],CB[9]);                 02416200 P  0654
                                                 GO TO STRT; END;                 %114-         02416300 C  0659
                    NEXTCARD ← SEQCHK(TB[9], CB[9]);                                            02417000 T  0660
               ENDP:                                                                            02418000 T  0662
                    IF NEXTCARD NEQ 7 THEN                                                      02418100 T  0663
                    IF VOIDTOG THEN IF SEQCHK(CRD[9],VOIDSEQ) < 4 THEN                          02419000 T  0663
                        VOIDTOG := FALSE ELSE GO TO STRT;                                       02419050 T  0668
                    IF GETCOL1(CRD[0]) THEN                                                     02419100 T  0670
               BEGIN                                                                            02419150 T  0671
                    BUF := MKABS(CRD[0]);                                                       02419200 T  0672
```

```
        BUF := SCANINC(BUF,ID,RESULT,1,0);                                          02419250 T  0674
        IF ID = INCLUDE THEN GO TO LIBADD ELSE GO TO STRT;                          02419300 T  0677
    END;                                                                            02419325 T  0679
        SEQERRORS := FALSE;                                                         02420000 T  0679
        IF NEXTCARD = 7 THEN                                                        02420010 T  0680
           BEGIN MOVESEQ(LASTSEQ,CRD[9]); GO TO ENDPA; END;                         02420011 T  0681
        IF CHECKTOG AND SEQERRCT=0 THEN SEQERRCT←1 ;                                02420015 T  0684
        IF CHECKTOG THEN IF SEQCHK(LASTSEQ,CRD[9])=3 THEN                           02420020 T  0687
           BEGIN                                                                    02420030 T  0690
           SEQERR(ERRORBUFF,CRD[9],LASTSEQ) ;                                       02420040 T  0691
           MOVESEQ(LASTSEQ,CRD[9]) ;                                                02420050 T  0693
           IF SEQTOG THEN                                                           02420053 T  0694
              BEGIN NEWSEQ(CRD[9],SEQBASE);SEQBASE←SEQBASE+SEQINCR END;             02420057 T  0695
           MOVESEQ(LINKLIST,CRD[9]);                                                02420060 T  0698
           MOVE(CRD[9],BUFL) ;                                                      02420070 T  0700
           SEQERRCT←SEQERRCT+1 ;                                                    02420080 T  0701
           SEQERRORS←TRUE ;                                                         02420090 T  0702
           IF NOT LISTOG THEN PRINTCARD ;                                           02420100 T  0704
           END                                                                      02420110 T  0706
        ELSE MOVESEQ(LASTSEQ,CRD[9]) ELSE MOVESEQ(LASTSEQ,CRD[9]) ;                 02420120 T  0706
                                                                                    02420150 T  0710
                                                                                    02420200 T  0710
                                                                                    02420250 T  0710
                                                                                    02420300 T  0710
                                                                                    02420350 T  0710
                                                                                    02420400 T  0710
                                                                                    02420450 T  0710
                                                                                    02420500 T  0710
ENDPA:                                                                              02420550 T  0710
        IF SEQTOG THEN                                                              02421000 T  0711
           BEGIN                                                                    02422000 T  0711
           NEWSEQ(CRD[9],SEQBASE);                                                  02423000 T  0712
           SEQBASE ← SEQBASE + SEQINCR;                                             02424000 T  0713
           END;                                                                     02425000 T  0714
     IF NOWRT THEN GO TO ENDPB;                                                     02425100 T  0714
     IF NEWTPTOG THEN IF TSSEDITOG THEN BEGIN TSSEDITS(CRD,PRINTBUFF) ;             02426000 T  0715
     WRITE(NEWTAPE,10,PRINTBUFF[*]) END ELSE WRITE(NEWTAPE,10,CRD[*]). ;            02426005 T  0719
ENDPB:                                                                              02426008 T  0728
     IF NOT SEQERRORS THEN                                                          02426010 T  0729
        BEGIN                                                                       02426020 T  0730
        MOVESEQ(LINKLIST,CRD[9]) ;                                                  02426030 T  0730
        MOVE(CRD[9],BUFL) ;                                                         02426040 T  0732
        END ;                                                                       02427000 T  0733
     NCR ← INITIALNCR;                                                              02428000 T  0733
     READACARD ← TRUE;                                                              02429000 T  0734
XIT:                                                                                02430000 T  0734
     SEGSWFIXED ← TRUE ;                                                            02431000 T  0735
     REMFIXED←TRUE ;                                                                02431005 T  0736
     IF TSSEDITOG THEN WARNED←TRUE ;                                                02431100 T  0738
     IF LISTOG AND FIRSTCALL THEN DATIME;                                           02432000 T  0741
     IF SEGSW THEN %%% ENTER SEQ# AND ADR TO LINESEG ARRAY,                         02432100 T  0744
        BEGIN IF LASTADDR≠ADR THEN BEGIN NOLIN←NOLIN+1;  LASTADDR←ADR END;          02432200 T  0745
        LINESEG[NOLIN,IR,NOLIN,IC]←0 & D2B(LASTSEQ)[10:20:28] & (ADR+3)             02432300 T  0748
        [38:36:10] ;                                                                02432400 T  0754
        END ;                                                                       02432500 T  0755
END READACARD;                                                                      02433000 T  0755
                                                                        80 IS  761 LONG, NEXT SEG    6
```

```
                INTEGER STREAM PROCEDURE CONVERT(NUB,SIZE,P,CHAR); VALUE P;                    02434000 T  0497
PRT(636) = CONVERT
                    BEGIN                                                                      02435000 T  0497
                    LOCAL T;                                                                   02436000 T  0498
                        SI ← P;                                                                02437000 T  0498
                        8(IF SC < "0" THEN JUMP OUT;                                           02438000 T  0498
                           SI ← SI + 1; TALLY ← TALLY + 1);                                    02439000 T  0499
                        CONVERT ← SI;                                                          02440000 T  0501
                        DI ← CHAR; DS ← 7 LIT "0"; DS ← CHR;                                   02441000 T  0501
                        T ← TALLY;                                                             02442000 T  0503
                        SI ← P; DI ← NUB; DS ← T OCT;                                          02443000 T  0503
                        DI ← SIZE ; SI ← LOC T; DS ← WDS;                                      02444000 T  0504
                    END CONVERT;                                                               02445000 T  0505
                PROCEDURE SCAN;                                                                02446000 T  0506
                BEGIN                                                                          02447000 T  0506
                BOOLEAN STREAM PROCEDURE ADVANCE(NCR, ACR, CHAR, NCRV, ACRV);                  02448000 T  0506
                                                                          START OF SEGMENT **********   81
PRT(637) = ADVANCE
                VALUE NCRV, ACRV, CHAR;                                                        02449000 T  0000
                BEGIN LABEL LOOP;                                                              02450000 T  0000
                   LABEL DIG, ALPH, BK1, BK2, SPEC;                                            02451000 T  0000
                   DI ← ACRV;                                                                  02452000 T  0000
                   SI ← CHAR; SI ← SI+8;                                                       02453000 T  0000
                   IF SC ≠ " " THEN                                                            02454000 T  0000
                      IF SC ≥ "0" THEN BEGIN SI ← NCRV; GO TO BK1 END ELSE                     02455000 T  0001
                                       BEGIN SI ← NCRV; GO TO BK2 END;                         02456000 T  0003
                   SI ← NCRV;                                                                  02457000 T  0004
                   LOOP;                                                                       02458000 T  0004
                   IF SC = " " THEN BEGIN SI←SI+1; GO TO LOOP END;                             02459000 T  0004
                   IF SC ≥ "0" THEN                                                            02460000 T  0006
                   BEGIN                                                                       02461000 T  0007
                      DIG: DS ← CHR;                                                           02462000 T  0007
                      BK1: IF SC = " " THEN BEGIN SI ← SI+1; GO TO BK1 END;                    02463000 T  0008
                      IF SC ≥ "0" THEN GO TO DIG;                                              02464000 T  0010
                      GO TO SPEC;                                                              02465000 T  0011
                   END;                                                                        02466000 T  0011
                   IF SC = ALPHA THEN                                                          02467000 T  0011
                   BEGIN                                                                       02468000 T  0012
                      ALPH: DS ← CHR;                                                          02469000 T  0012
                      BK2: IF SC = " " THEN BEGIN SI ← SI+1; GO TO BK2 END;                    02470000 T  0013
                      IF SC = ALPHA THEN GO TO ALPH;                                           02471000 T  0015
                   END;                                                                        02472000 T  0016
                   SPEC:                                                                       02473000 T  0016
                   ACRV ← DI;                                                                  02474000 T  0016
                   DS ← 6 LIT " ";                                                             02475000 T  0017
                   IF SC = "]" THEN                                                            02476000 T  0018
                   BEGIN TALLY ← 1; SI ← LOC ACRV;                                             02477000 T  0018
                      DI ← ACR; DS ← WDS;                                                      02478000 T  0019
                      DI ← CHAR; DS ← LIT ";";                                                 02479000 T  0020
                   END ELSE                                                                    02480000 T  0021
                   BEGIN DI ← CHAR; DS ← CHR;                                                  02481000 T  0021
                      ACRV ← SI; SI ← LOC ACRV;                                                02482000 T  0021
                      DI ← NCR; DS ← WDS;                                                      02483000 T  0022
                   END;                                                                        02484000 T  0022
                   ADVANCE ← TALLY;                                                            02485000 T  0022
                END ADVANCE;                                                                   02486000 T  0023
                BOOLEAN PROCEDURE CONTINUE;                                                    02487000 T  0024
```

```
PRT(640) = CONTINUE
                BEGIN
                LABEL LOOP;

                BOOLEAN STREAM PROCEDURE CONTIN(CD);
PRT(641) = CONTIN
                BEGIN SI←CD;  IF SC≠"C" THEN IF SC≠"$" THEN BEGIN SI←SI+5;  IF SC≠" "
                THEN IF SC≠"0" THEN BEGIN TALLY←1;  CONTIN←TALLY END END END OF CONTIN ;
                BOOLEAN STREAM PROCEDURE COMNT(CD,T);  VALUE T ;
PRT(642) = COMNT
                BEGIN LABEL L ;
                   SI ← CD; IF SC = "C" THEN BEGIN T(SI←SI+1;  IF SC="-" THEN TALLY←1
                   ELSE JUMP OUT TO L);  TALLY←1;  L: END;  COMNT←TALLY ;
                END COMNT;
                BOOLEAN STREAM PROCEDURE DCCONTIN(CD);
PRT(643) = DCCONTIN
                BEGIN
                    SI ← CD; IF SC = "-" THEN TALLY ← 1;
                    DCCONTIN ← TALLY;
                END DCCONTIN;
                LOOP:    IF NOT(CONTINUE ←
                           IF(DCINPUT AND NOT TSSEDITOG)OR FREEFTOG THEN
                              IF NEXTCARD < 4 THEN DCCONTIN(CB)
                              ELSE IF NEXTCARD = 7 THEN DCCONTIN(DB)ELSE CONTIN(TB)
                           ELSE IF NEXTCARD = 7 THEN CONTIN(DB)
                           ELSE IF NEXTCARD < 4 THEN CONTIN(CB) ELSE
                           CONTIN(TB)) THEN
                         IF(IF NEXTCARD < 4 THEN
                              COMNT(CB,(DCINPUT AND NOT TSSEDITOG) OR FREEFTOG)
                            ELSE IF NEXTCARD = 7 THEN
                              COMNT(DB,(DCINPUT AND NOT TSSEDITOG) OR FREEFTOG)
                         ELSE COMNT(TB,0) AND NEXTCARD ≠ 6) THEN
                         BEGIN
                           IF READACARD THEN IF LISTOG THEN PRINTCARD;
                            GO TO LOOP;
                         END;
                END CONTINUE;

                PROCEDURE SCANX(EOF1, EOF2, EOS1, EOS2, OK1, OK2);
PRT(644) = SCANX
                   VALUE       EOF1, EOF2, EOS1, EOS2, OK1, OK2;
                   INTEGER     EOF1, EOF2, EOS1, EOS2, OK1, OK2;
                BEGIN LABEL LOOP, LOOP0 ;

                   LOOP0:
                   EXACCUM[1] ← BLANKS;
                   ACR ← ACR1;
                   LOOP:
                   IF ADVANCE(NCR, ACR, CHR1, NCR, ACR) THEN
                      IF CONTINUE THEN
                %
                         IF READACARD THEN
                              BEGIN
                              IF LISTOG THEN PRINTCARD ;
                              IF ACR,[33:15]≥EXACCUMSTOP THEN
                                   BEGIN XTA←BLANKS; FLOG(175); GO LOOP0 END ;
                              GO LOOP ;
```

```
                                                                    02487200 T  0024
                                                                    02487400 T  0024
                                                  START OF SEGMENT **********   82
                                                                    02487600 T  0000
                                                                    02488000 T  0000
                                                                    02489000 T  0002
                                                                    02489200 T  0005
                                                                    02489400 T  0006
                                                                    02489600 T  0006
                                                                    02490000 T  0009
                                                                    02490200 T  0011
                                                                    02490400 T  0012
                                                                    02490500 T  0012
                                                                    02490600 T  0013
                                                                    02491000 T  0014
                                                                    02491200 T  0014
                                                                    02491400 T  0015
                                                                    02491600 T  0016
                                                                    02491650 T  0019
                                                                    02491700 T  0021
                                                                    02491800 T  0026
                                                                    02492000 T  0030
                                                                    02492200 T  0034
                                                                    02492400 T  0036
                                                                    02492450 T  0038
                                                                    02492500 T  0041
                                                                    02492550 T  0044
                                                                    02492600 T  0048
                                                                    02493000 T  0052
                                                                    02493200 T  0053
                                                                    02493400 T  0055
                                                                    02493600 T  0056
                                                                    02494000 T  0056
                                                    82 IS   59 LONG, NEXT SEG   81
                                                                    02495000 T  0024

                                                                    02496000 T  0024
                                                                    02497000 T  0024
                                                                    02498000 T  0024
                                                  START OF SEGMENT **********   83
                                                                    02498100 T  0000
                                                                    02499000 T  0000
                                                                    02500000 T  0001
                                                                    02501000 T  0002
                                                                    02502000 T  0002
                                                                    02503000 T  0004
                                                                    02504000 T  0005
                                                                    02505000 T  0005
                                                                    02506000 T  0006
                                                                    02506010 T  0006
                                                                    02506020 T  0008
                                                                    02506030 T  0009
                                                                    02506040 T  0012
```

```
                        END
                    ELSE SCN ← IF EXACCUM[1] = BLANKS THEN EOF1 ELSE EOF2
                ELSE SCN ← IF EXACCUM[1] = BLANKS THEN EOS1 ELSE EOS2
            ELSE SCN ← IF EXACCUM[1] = BLANKS THEN OK1 ELSE OK2;
        END SCANX;

        DEFINE CHAR = ACCUM[0]#;
        DEFINE T=SYMBOL#;
        INTEGER N;
STACK(F+2) = N
        BOOLEAN STREAM PROCEDURE CHECKEXP(NCR, NCRV, A); VALUE NCRV;
PRT(645) = CHECKEXP
        BEGIN
            SI ← NCRV;
            IF SC = "*" THEN
            BEGIN DI ← A; DI ← DI+2; DS ← 2 LIT "*"; SI ← SI+1; NCRV ← SI;
                TALLY ← 1; CHECKEXP ← TALLY;
                SI ← LOC NCRV; DI ← NCR; DS ← WDS  END;
        END CHECKEXP;
        PROCEDURE CHECKRESERVED;
PRT(646) = CHECKRESERVED
        BEGIN LABEL RESWD, XIT, FOUND1, FOUND2, DONE;

        BOOLEAN STREAM PROCEDURE COMPLETECHECK(A,B,N);  VALUE N ;
PRT(647) = COMPLETECHECK
            BEGIN LABEL L ;
            SI←A; SI←SI+2; DI←B; N(IF SC≠DC THEN JUMP OUT TO L); TALLY←1;
            L: COMPLETECHECK←TALLY ;
            END OF COMPLETECHECK ;
        STREAM PROCEDURE XFER(FROM, T1, T2, N, M); VALUE FROM, N, M;
PRT(650) = XFER
        BEGIN SI ← FROM; DI ← T1; DI ← DI+2;
            DS ← M CHR;
            SI ← FROM; SI ← SI+N;
            DI ← T2; DI ← DI+2;
            DS ← 6 CHR;
        END XFER;
        STREAM PROCEDURE XFERA(FROM, NEXT1, NEXT2);
PRT(651) = XFERA
            VALUE FROM;
        BEGIN SI ← FROM; SI ← SI+6;
            DI ← NEXT1; DI ← DI+2;
            5(IF SC ≥ "0" THEN DS ← CHR ELSE JUMP OUT);
            SI ← SI+2;
            DI ← NEXT2; DI ← DI+2;
            6(IF SC = ALPHA THEN DS ← CHR ELSE JUMP OUT);
        END XFERA;
        BOOLEAN STREAM PROCEDURE CHECKFUN(FROM, TOO, N); VALUE FROM, N;
PRT(652) = CHECKFUN
        BEGIN SI ← FROM; SI ← SI + N;
            IF SC = "0" THEN
            BEGIN SI ← SI+1;
                IF SC = "N" THEN
                BEGIN SI ← SI+1; TALLY ← 1;
                    DI ← TOO; DI ← DI+2;
                    DS ← 6 CHR;
                END;
```

```
02506050 T  0012
02507000 T  0012
02508000 T  0015
02509000 T  0019
02510000 T  0023
83 IS   24 LONG, NEXT SEG   81
02511000 T  0024
02512000 T  0024
02513000 T  0024

02514000 T  0024

02515000 T  0024
02516000 T  0024
02517000 T  0024
02518000 T  0024
02519000 T  0026
02520000 T  0027
02521000 T  0028
02522000 T  0029

02523000 T  0029
START OF SEGMENT **********   84
02523100 T  0000

02523200 T  0000
02523300 T  0000
02523400 T  0003
02523500 T  0004
02524000 T  0005

02525000 T  0005
02526000 T  0006
02527000 T  0007
02528000 T  0008
02529000 T  0008
02530000 T  0008
02531000 T  0009

02532000 T  0009
02533000 T  0009
02534000 T  0009
02535000 T  0010
02536000 T  0012
02537000 T  0012
02538000 T  0012
02539000 T  0015
02540000 T  0015

02541000 T  0015
02542000 T  0016
02543000 T  0017
02544000 T  0018
02545000 T  0018
02546000 T  0019
02547000 T  0020
02548000 T  0020
```

```
                    END;                                                                    02549000 T  0020
                    CHECKFUN ← TALLY;                                                       02550000 T  0020
                END CHECKFUN;                                                               02551000 T  0020
                BOOLEAN STREAM PROCEDURE MORETHAN6(P);                                      02552000 T  0021
PRT(653) = MORETHAN6
                BEGIN S1 ← P;                                                               02553000 T  0021
                    IF SC ≠ " " THEN TALLY ← 1;                                             02554000 T  0022
                    MORETHAN6 ← TALLY;                                                      02555000 T  0023
                END MORETHAN6;                                                              02556000 T  0023
                INTEGER I; ALPHA ID;                                                        02557000 T  0024
STACK(F+2) = I
STACK(F+3) = ID
                INTEGER STOR ;                                                              02557100 T  0024
STACK(F+4) = STOR
                    IF ACCUM[1] = "      " THEN                                             02558000 T  0024
                    BEGIN XTA ← CHAR; FLOG(16); GO TO XIT END;                              02559000 T  0026
                    IF CHAR = "=     " OR CHAR = "#     " THEN GO TO XIT;                   02560000 T  0030
                    IF CHAR = "←     " THEN GO TO XIT;                                      02560100 T  0033
                    IF CHAR ≠ "(     " AND CHAR ≠ "%     " THEN GO TO RESWD;                02561000 T  0034
                    IF MORETHAN6(ACCUM[2]) THEN GO TO RESWD;                                02562000 T  0037
                    COMMENT  AT THIS POINT WE HAVE <ID> ( .                                 02563000 T  0039
                    THIS MUST BE ONE OF THE FOLLOWING;                                      02564000 T  0039
                    ASSIGNEMNT STATEMENT WITH SUBSCRIPTED VARIABLE AT THE LEFT,             02565000 T  0039
                    STATEMENT FUNCTION DECLARATION,                                         02566000 T  0039
                CALL, REAL, ENTRY, GO TO, READ, WRITE, FORMAT, IF, DATA, CHAIN, PRINT OR    02567000 T  0039
                  PUNCH;                                                                    02567100 T  0039
                    IF I ← SEARCH(T) > 0 THEN                                               02568000 T  0039
                         IF GET(I).CLASS = ARRAYID THEN GO TO XIT;                          02569000 T  0041
                    ID ← T; ID.[36:12] ← "  ";                                              02570000 T  0044
                     FOR I←0 THRU RSP DO IF RESERVEDWORDSLP[I]=ID THEN IF (IF STOR          02571000 T  0046
                     ←RESLENGTHLP[I]-4<1 THEN TRUE ELSE COMPLETECHECK(ACCUM[2],             02571100 T  0055
                     RESERVEDWORDSLP[I+RSP1],STOR)) THEN GO FOUND1 ;                        02572000 T  0060
                    GO TO XIT;                                                              02573000 T  0064
                    FOUND1;                                                                 02574000 T  0065
                    NEXT ← LPGLOBAL[I];                                                     02575000 T  0065
                    T ← "      ";                                                           02576000 T  0066
                    XFER(ACRO, T, NEXTACC, I←RESLENGTHLP[I], IF I> 6 THEN 6 ELSE I);        02577000 T  0066
                    GO TO DONE;                                                             02578000 T  0071
                    RESWD;                                                                  02579000 T  0073
                    COMMENT  AT THIS POINT WE KNOW THE <ID> MUST BE A SPECIAL WORD          02580000 T  0073
                    TO IDENTIFY THE STATEMENT TYPE;                                         02581000 T  0073
                    ID ← T; ID.[36:12] ← "  ";                                              02582000 T  0073
                    IF T = "ASSIGN" THEN                                                    02583000 T  0075
                    BEGIN                                                                   02584000 T  0076
                       NEXTSCN ← SCN; SCN ← 14;                                             02585000 T  0076
                       NEXTACC ← NEXTACC2 ← "      ";                                       02586000 T  0078
                       XFERA(ACRO, NEXTACC, NEXTACC2);                                      02587000 T  0079
                       NEXT ← 1;                                                            02588000 T  0080
                       GO TO XIT;                                                           02589000 T  0081
                    END;                                                                    02590000 T  0085
                     FOR I←1 THRU RSH DO IF RESERVEDWORDS[I]=ID THEN IF (IF STOR←           02591000 T  0085
                     RESLENGTH[I]-4<1 THEN TRUE ELSE COMPLETECHECK(ACCUM[2],RESERVEDWORDS   02591100 T  0087
                     [I+RSH1],IF STOR>8 THEN 8 ELSE STOR)) THEN GO FOUND2 ;                 02592000 T  0092
                    XTA ← T; FLOG(16); GO TO XIT;                                           02593000 T  0098
                    FOUND2;                                                                 02594000 T  0100
                    NEXT ← I+1;                                                             02595000 T  0101
                    T ← "      ";                                                           02596000 T  0102
```

```
                    XFER(ACRO, T, NEXTACC, I←RESLENGTH  [I], IF I> 6 THEN 6 ELSE I);                02597000 T  0103
                    DONE; NEXTSCN ← SCN;                                                            02598000 T  0107
                    SCN ← 6;                                                                        02599000 T  0108
                    IF NEXTACC = "FUNCTI" THEN                                                      02600000 T  0109
                       IF CHECKFUN(ACRO, NEXTACC, I+6) THEN SCN ← 13;                               02601000 T  0110
                    XIT:                                                                            02602000 T  0114
                    EOSTOG←FALSE;                                                                   02603000 T  0115
                 END CHECKRESERVED;                                                                 02604000 T  0115
                                                                                          84 IS  120 LONG, NEXT SEG   81
                 BOOLEAN PROCEDURE CHECKOCTAL;                                                      02605000 T  0029
PRT(654) = CHECKOCTAL
                 BEGIN                                                                              02606000 T  0029
                    INTEGER S, T; LABEL XIT;                                                        02607000 T  0029
                                                                                       START OF SEGMENT **********   85
STACK(F+3) = S
STACK(F+4) = T
                 INTEGER STREAM PROCEDURE COUNT(ACRV,T);   VALUE ACRV,T ;                           02608000 T  0000
PRT(655) = COUNT
                    BEGIN                                                                           02609000 T  0000
                    LOCAL A,B;   SI←LOC T;   SI←SI+7 ;                                              02610000 T  0000
                    IF SC="1" THEN BEGIN SI←ACRV;IF SC="0" THEN SI←SI+1 END ELSE SI←ACRV;           02611000 T  0000
                    IF SC≠" " THEN                                                                  02612000 T  0003
                       BEGIN  A←SI ;                                                                02613000 T  0003
                       17(IF SC>"7" THEN BEGIN TALLY←17; JUMP OUT END ELSE IF SC<"0" THEN            02614000 T  0004
                       BEGIN IF SC≠" " THEN TALLY←17; JUMP OUT END; SI←SI+1;                         02614010 T  0006
                       TALLY←TALLY+1) ;                                                             02614015 T  0009
                       B←TALLY;   SI←LOC B;   SI←SI+7 ;                                             02614020 T  0010
                       IF SC="+" THEN BEGIN SI←A; IF SC>"3" THEN TALLY←17 END;                      02614030 T  0010
                       END ;                                                                        02614040 T  0012
                    COUNT←TALLY ;                                                                   02614050 T  0012
                    END OF COUNT ;                                                                  02614060 T  0013
                 ALPHA STREAM PROCEDURE CONV(ACRV, S, T); VALUE ACRV, S, T;                         02615000 T  0014
PRT(656) = CONV
                 BEGIN S↑ ← ACRV; IF SC = "0" THEN S↑ ← SI+1;                                       02616000 T  0014
                    DI ← LOC CONV; SKIP S DB;                                                       02617000 T  0015
                    T(SKIP 3 SB; 3(IF SB THEN DS ← SET ELSE DS ← RESET; SKIP 1 SB));                02618000 T  0015
                 END CONV;                                                                          02619000 T  0019
                    IF T←COUNT(ACRO,1) = 0 THEN                                                     02620000 T  0020
                    BEGIN S ← 1;                                                                    02620100 T  0024
                       IF T ← CHAR ≠ "+      " AND T ≠ "&      " THEN                               02620200 T  0025
                       IF T = "-      " THEN S ← -1 ELSE GO TO XIT;                                 02621000 T  0027
                       SCANX(4, 4, 3, 3, 10, 10);                                                   02621100 T  0030
                       IF SCN ≠ 10 THEN GO TO XIT;                                                  02622000 T  0032
                       IF T←COUNT(ACR1,2) = 0 OR T > 16 THEN GO TO XIT ;                            02622100 T  0033
                       FNEXT ← CONV(ACR1, (16-T)×3, T);                                             02622200 T  0038
                       IF S < 0 THEN FNEXT ← -FNEXT;                                                02623000 T  0041
                    END ELSE IF T < 17 THEN FNEXT←CONV(ACRO,(16-T)×3,T) ELSE GO TO XIT ;            02623100 T  0044
                    CHECKOCTAL ← TRUE;                                                              02624000 T  0052
                    NEXT ← NUM;                                                                     02625000 T  0053
                    NUMTYPE ← REALTYPE;                                                             02626000 T  0054
                    XIT:                                                                            02627000 T  0054
                 END CHECKOCTAL;                                                                    02628000 T  0055
                                                                                          85 IS   58 LONG, NEXT SEG   81
                 PROCEDURE HOLLERITH;                                                               02629000 T  0029
PRT(657) = HOLLERITH
                 BEGIN                                                                              02630000 T  0029
                    REAL T, COL1, T2, ENDP;                                                         02631000 T  0029
```

```
STACK(F+2) = T
STACK(F+3) = COL1
STACK(F+4) = T2
STACK(F+5) = ENDP
                    LABEL XIT;
                    INTEGER STREAM PROCEDURE STRCNT(S,D,SZ); VALUE S,SZ;                    02632000 T  0000
                                                                                            02633000 T  0000
PRT(660) = STRCNT
                     BEGIN                                                                  02634000 T  0000
                          SI ← S; DI ← D;DS ← 8 LIT "00      "; DI ← D;                   02635000 T  0000
                          DI ← D; DI ← DI + 2; DS ←SZ CHR; STRCNT ← SI;                     02636000 T  0002
                     END STRCNT;                                                            02637000 T  0003
                    INTEGER STREAM PROCEDURE RSTORE(S,D,SKP,SZ);                            02638000 T  0004
PRT(661) = RSTORE
                          VALUE S, SKP, SZ;                                                 02639000 T  0004
                     BEGIN                                                                  02640000 T  0004
                          DI ← D;                                                           02641000 T  0005
                          SI ← S; DI ←DI + SKP; DS ← SZ CHR; RSTORE ← SI;                   02642000 T  0005
                     END RSTORE;                                                            02643000 T  0006
                    F1 ← FNEXT;                                                             02644000 T  0007
                    NUMTYPE ← STRINGTYPE;                                                   02645000 T  0008
                    T ← 0 & NCR[30:33:15] & NCR[45:30:3];                                   02646000 T  0009
                    COL1 ← 0 & INITIALNCR[30:33:15];                                        02647000 T  0012
                    ENDP ← COL1 + 72;                                                       02648000 T  0014
                    STRINGSIZE ← 0;                                                         02649000 T  0015
                    WHILE F1 >0 DO                                                          02650000 T  0016
                    BEGIN                                                                   02651000 T  0017
                     T2 ← IF F1 > 6 THEN 6 ELSE F1;                                         02652000 T  0017
                       IF STRINGSIZE > MAXSTRING THEN                                       02653000 T  0020
                          BEGIN FLAG(120); STRINGSIZE ← 0 END;                              02654000 T  0020
                     IF T+T2> ENDP THEN IF DCINPUT OR FREEFTOG THEN                         02655000 T  0022
                          BEGIN XTA←BLANKS;  FLOG(150);  GO TO XIT END                      02655500 T  0026
                     ELSE BEGIN                                                             02656000 T  0028
                       IF TSSEDITOG THEN IF NOT DCINPUT THEN TSSED(BLANKS,1) ;              02656100 T  0029
                       NCR ← STRCNT(NCR, STRINGARRAY[STRINGSIZE], ENDP-T);                  02657000 T  0033
                       IF NOT CONTINUE THEN                                                 02658000 T  0036
%                                                                                           02659000 T  0036
                          BEGIN FLOG(43); GO TO XIT END;                                    02660000 T  0036
                       IF READACARD THEN;                                                   02661000 T  0038
                       IF LISTOG THEN PRINTCARD;                                            02662000 T  0039
                       NCR ← RSTORE(NCR,STRINGARRAY[STRINGSIZE],ENDP-T+2,T2-(ENDP-T));      02663000 T  0041
                       STRINGSIZE ← STRINGSIZE+1;                                           02664000 T  0046
                       F1 ← F1 - T2;                                                        02665000 T  0047
                       T ← COL1 + 6 + T2 - (ENDP - T);                                      02666000 T  0048
                    END ELSE                                                                02667000 T  0051
                    BEGIN                                                                   02668000 T  0051
                       NCR ← STRCNT(NCR, STRINGARRAY[STRINGSIZE], T2);                      02669000 T  0051
                       STRINGSIZE ← STRINGSIZE +1;                                          02670000 T  0054
                       T ← T +T2;                                                           02671000 T  0055
                       F1 ← F1 - T2;                                                        02672000 T  0056
                    END;                                                                    02673000 T  0058
                    END;                                                                    02674000 T  0058
                    NUMTYPE ← STRINGTYPE;                                                   02675000 T  0058
                    SCN ← 1;                                                                02676000 T  0059
                    XIT:                                                                    02677000 T  0060
END HOLLERITH;                                                                              02678000 T  0060
```

```
                    PROCEDURE QUOTESTRING;                                                                  02679000 T  0029
PRT(662) = QUOTESTRING
                    BEGIN                                                                                   02680000 T  0029
                       REAL C;                                                                              02681000 T  0029
                                                                                                  START OF SEGMENT *********    87
STACK(F+2) = C
                       LABEL XIT;                                                                           02682000 T  0000
                       ALPHA STREAM PROCEDURE STRINGWORD(S,D,SKP,SZ,C);                                     02683000 T  0000
PRT(663) = STRINGWORD
                       VALUE S,SKP,SZ;                                                                      02684000 T  0000
                       BEGIN                                                                                02685000 T  0000
                          LABEL QT, XIT;                                                                    02686000 T  0000
                          DI ← D; SI ← S;                                                                   02687000 T  0000
                          DI ← DI+SKP; DI ← DI+2;                                                           02688000 T  0000
                          TALLY ← SKP;                                                                      02689000 T  0001
                          SZ( IF SC = """ THEN JUMP OUT TO QT;                                              02690000 T  0002
                               IF SC = ":" THEN JUMP OUT TO QT;                                             02691000 T  0004
                               IF SC = "@" THEN JUMP OUT TO QT;                                             02692000 T  0005
                               IF SC = "]" THEN JUMP OUT TO XIT;                                            02693000 T  0006
                              DS ← CHR; TALLY ← TALLY+1);                                                   02694000 T  0007
                          GO TO XIT;                                                                        02695000 T  0007
                          QT: TALLY ← TALLY+7; SI ← SI+1;                                                   02696000 T  0008
                          XIT: STRINGWORD ← SI; S ← TALLY;                                                  02697000 T  0008
                          SI ← LOC S; DI ← C; DS ← WDS;                                                     02698000 T  0009
                       END STRINGWORD;                                                                      02699000 T  0010
                       STRINGSIZE ← 0;                                                                      02700000 T  0011
                       DO                                                                                   02701000 T  0012
                          BEGIN                                                                             02702000 T  0013
                          IF STRINGSIZE > MAXSTRING THEN                                                    02703000 T  0013
                          BEGIN FLAG(120); STRINGSIZE ← 0 END;                                              02704000 T  0013
                          STRINGARRAY[STRINGSIZE] ← BLANKS;                                                 02705000 T  0015
                          NCR ← STRINGWORD(NCR, STRINGARRAY[STRINGSIZE], 0, 6, C);                          02706000 T  0017
                          IF C<6 THEN IF DCINPUT OR FREEFTOG                                                02707000 T  0020
                                      THEN BEGIN XTA←BLANKS; FLOG(150); GO TO XIT END                       02707500 T  0022
                          ELSE BEGIN                                                                        02708000 T  0025
                          IF TSSEDITOG THEN IF NOT DCINPUT THEN TSSED(BLANKS,1) ;                           02708100 T  0026
                          IF NOT CONTINUE THEN                                                              02709000 T  0029
                    %                                                                                       02710000 T  0030
                             BEGIN FLOG(121); GO TO XIT END;                                                02711000 T  0030
                             IF READACARD THEN;                                                             02712000 T  0032
                             IF LISTOG THEN PRINTCARD;                                                      02713000 T  0033
                             NCR ← STRINGWORD(NCR, STRINGARRAY[STRINGSIZE  ],C,6-C,C);                      02714000 T  0035
                          END;                                                                              02715000 T  0038
                          STRINGSIZE ← STRINGSIZE + 1;                                                      02716000 T  0038
                       END UNTIL C ≥ 7;                                                                     02717000 T  0039
                       IF C = 7 THEN STRINGSIZE ← STRINGSIZE-1;                                             02718000 T  0041
                       FNEXT ← STRINGSIZE;                                                                  02719000 T  0043
                       NEXT ← NUM;                                                                          02720000 T  0044
                       SYMBOL ← NAME ← STRINGARRAY[0];                                                      02721000 T  0045
                       NUMTYPE ← STRINGTYPE;                                                                02722000 T  0046
                       SCN ← 1;                                                                             02723000 T  0047
                       XIT:                                                                                 02724000 T  0048
                    END QUOTESTRING;                                                                        02725000 T  0048
                                                                                                  87 IS   51 LONG, NEXT SEG   81
                    PROCEDURE CHECKPERIOD;                                                                  02726000 T  0029
PRT(664) = CHECKPERIOD
                    BEGIN                                                                                   02727000 T  0029
```

```
                    LABEL FRACTION, XIT, EXPONENT, EXPONENTSIGN;                                    02728000 T  0029
                                                                                          START OF SEGMENT **********    88
                    LABEL NUMFINI, FPLP, CHKEXP;                                                    02729000 T  0000
                    ALPHA S, T, I, TS;                                                              02730000 T  0000
STACK(F+2) = S
STACK(F+3) = T
STACK(F+4) = I
STACK(F+5) = TS

                     INTEGER C2;                                                                    02730050 T  0000
STACK(F+6) = C2

                    BOOLEAN CON;                                                                    02730100 T  0000
STACK(F+7) = CON

                     IF T ← CHAR ≠ ",      " THEN GO TO CHKEXP;                                     02731000 T  0000
                    SCANX(4, 9, 3, 8, 10, 11);                                                      02732000 T  0002
                    IF T ← EXACCUM[1] = "       " THEN                                              02733000 T  0004
                      BEGIN IF NUMTYPE ≠ DOUBTYPE THEN NUMTYPE ← REALTYPE; GO TO XIT END;           02733500 T  0005
                    IF T = "E     " OR T = "D     " THEN GO TO EXPONENTSIGN;                        02734000 T  0011
                    IF T.[12:6] ≤ 9 THEN GO TO FRACTION;                                            02735000 T  0013
                    IF T.[18:6] ≤ 9 THEN                                                            02736000 T  0015
                    BEGIN                                                                           02737000 T  0016
                       IF S ← T.[12:6] ≠ "E" AND S ≠ "D" THEN                                       02738000 T  0017
                          BEGIN XTA ← T; FLOG(63); GO TO XIT END;                                   02739000 T  0019
                       EXACCUM[1].[12:6] ← 0;                                                       02740000 T  0025
                       I ← 1; GO TO EXPONENT;                                                       02741000 T  0027
                    END;                                                                            02742000 T  0028
                    IF EXACCUM[0] ≠ ",      " THEN GO TO XIT;                                       02743000 T  0028
                    FOR I ← 0 STEP 1 UNTIL 10 DO                                                    02744000 T  0030
                       IF T = PERIODWORD[I] THEN                                                    02745000 T  0032
                       BEGIN EXACCUM[2] ← I; SCN ← 12; GO TO XIT END;                               02746000 T  0033
                    GO TO XIT;                                                                      02747000 T  0038
                    FRACTION: NEXT ← NUM;                                                           02748000 T  0038
                    IF NUMTYPE ≠DOUBTYPE THEN NUMTYPE ← REALTYPE;  XTA ← ACR1;                      02749000 T  0039
                    FPLP:                                                                           02750000 T  0042
                    F1 ← 0;                                                                         02751000 T  0043
                    XTA  ← CONVERT(F1,C1,XTA ,TS);                                                  02752000 T  0043
                    C2 ← C2 + C1;                                                                   02753000 T  0046
                    IF (F2 ← FNEXT×TEN[C1]+F1) ≤ MAX                                                02754000 T  0047
                     THEN FNEXT ← F2                                                                02755000 T  0049
                     ELSE BEGIN                                                                     02756000 T  0050
                          NUMTYPE ← DOUBTYPE;                                                       02757000 T  0052
                           CON ← TRUE;                                                              02757100 T  0052
                          DOUBLE(FNEXT,DBLOW,TEN[C1],TEN[69+C1],×,                                  02758000 T  0053
                                F1,0,+,←,FNEXT,DBLOW);                                              02759000 T  0056
                          END;                                                                      02760000 T  0057
                    IF TS ≤ 9 THEN GO TO FPLP;                                                      02761000 T  0057
                    F1 ← 0;                                                                         02762000 T  0059
                    IF T ← EXACCUM[0] ≠ "E     " AND T ≠ "D     " THEN                               02763000 T  0059
                    BEGIN IF SCN = 8 THEN SCN ← 3 ELSE SCN ← 10;                                    02764000 T  0062
                       GO TO NUMFINI;                                                               02765000 T  0068
                    END;                                                                            02766000 T  0069
                    CHKEXP:  FNEXT ← FNEXT × 1.0;                                                   02767000 T  0069
                    F1 ←0;                                                                          02768000 T  0071
                    I ← 1;                                                                          02769000 T  0072
                    SCANX(4, 4, 3, 3, 20, 10);                                                      02770000 T  0072
                    IF SCN = 20 THEN                                                                02771000 T  0074
                    EXPONENTSIGN;                                                                   02772000 T  0075
                    BEGIN IF S ← EXACCUM[0] ≠ "+     " AND S ≠ "&     " THEN                       02773000 T  0076
```

```
      IF S = "-      " THEN I ← -1 ELSE                                                    02774000 T  0078
         BEGIN XTA ← S; FLOG(63); SCN ← 10; GO TO XIT END;                                  02775000 T  0081
      SCANX(4, 4, 3, 3, 10, 10);                                                           02776000 T  0088
   END;                                                                                    02777000 T  0090
   IF (S ← EXACCUM[1]).[12:6] > 9 THEN                                                     02778000 T  0090
   BEGIN XTA ← IF S ≠ BLANKS THEN S ELSE T; FLOG(63); GO TO XIT END;                        02778100 T  0092
   EXPONENT:                                                                               02779000 T  0097
   IF NUMTYPE ≠ DOUBTYPE THEN NUMTYPE ← REALTYPE;                                          02779500 T  0098
IF T.[12:6] = "D" THEN NUMTYPE ← DOUBTYPE;                                                 02780000 T  0100
   IF SCN = 8 THEN SCN ← 3 ELSE IF SCN = 11 THEN SCN ← 10;                                 02781000 T  0102
   XTA ← ACR1;                                                                             02782000 T  0107
   XTA  ← CONVERT(F1,C1,XTA ,TS);                                                          02783000 T  0107
   IF I < 0 THEN F1 ← -F1;                                                                 02784000 T  0110
   NUMFINI:                                                                                02785000 T  0112
   C1 ← F1 - C2;                                                                           02786000 T  0113
   IF I ← (ABS(C1+(FNEXT.[3:6]&FNEXT[1:2:1]))) > 63 OR((ABS(C1) = I OR                     02787000 T  0114
       FNEXT ≥ 5) AND ABS(F1) ≥ 69)                                                        02787500 T  0118
    THEN BEGIN XTA ← T; FLOG(87); GO TO XIT; END;                                          02788000 T  0120
   IF NUMTYPE ≠ DOUBTYPE THEN                                                              02789000 T  0123
   BEGIN                                                                                   02790000 T  0124
      IF C1≥ 0 THEN FNEXT ← FNEXT × TEN[C1]                                                02791000 T  0125
               ELSE FNEXT ← FNEXT / TEN[-C1];                                              02792000 T  0126
   END ELSE                                                                                02793000 T  0130
   BEGIN                                                                                   02794000 T  0130
      IF C1 ≥ 0                                                                            02795000 T  0130
         THEN DOUBLE(FNEXT,DBLOW,TEN[C1],TEN[69+C1],×,←,FNEXT,DBLOW)                       02796000 T  0130
         ELSE DOUBLE(FNEXT,DBLOW,TEN[-C1],TEN[69-C1],/,←,FNEXT,DBLOW);                     02797000 T  0135
      IF CON THEN IF DBLOW.[9:33] = MAX.[9:33] THEN                                        02797100 T  0139
       IF FNEXT.[3:6] LSS 14                                                               02797150 T  0142
         THEN IF BOOLEAN(FNEXT.[2:1]) THEN                                                 02797200 T  0143
                  BEGIN DBLOW := 0; FNEXT := FNEXT + 1&FNEXT[2:2:7]; END;                  02797300 T  0145
      END;                                                                                 02798000 T  0148
      XIT:                                                                                 02799000 T  0148
   END CHECKPERIOD;                                                                        02800000 T  0149
                                                                                  88 IS  153 LONG, NEXT SEG   81
   LABEL LOOPO, NUMBER ;                                                                   02801000 T  0029
   LABEL L,XIT;                                                                            02802000 T  0029
   LABEL L1,L2,L3,L4,L5,L6,L7,L8,L9,L10,L11,L12,L13,L14,L15,L16,L17,                        02802100 T  0029
         L18,L19,L20,L21,BK ;                                                              02802200 T  0029
   SWITCH CASEL←L1,L2,L3,L4,L5,L6,L7,L8,L9,L10,L11,L12,L13,L14,L15,                        02802300 T  0029
                L16,L17,L18,L19,L20,L21 ;                                                  02802400 T  0031
   LABEL LOOP, CASESTMT;                                                    %994-         02803000 P  0042
   LABEL CASE0,CASE1,CASE2,CASE3,CASE4,CASE5,CASE6,CASE7;                   %994-         02803100 C  0042
   LABEL CASE8,CASE9,CASE10,CASE11,CASE12,CASE13,CASE14;                    %994-         02803200 C  0042
   PREC←NEXT←FNEXT←REAL(SCANENTER←FALSE) ;                                               02804000 T  0042
   CASESTMT:                                                                             02805000 T  0045
   CASE SCN OF                                                                           02806000 T  0046
   BEGIN                                                                                 02807000 T  0046
PRT(665) = *CASE STATEMENT DESCRIPTOR*
   CASE0:                                                                   %994-         02807999 C  0046
   GO TO IF LABELR THEN CASE5 ELSE CASE1;                                                02808000 T  0047
PRT(666) = CASE5
PRT(667) = CASE1
   CASE1:                                                                                02809000 T  0052
   BEGIN                                                                                 02810000 T  0053
      LOOPO:                                                                             02810100 T  0053
      ACR ← ACR0;                                                                        02811000 T  0053
```

```
   ACCUM[1] ← BLANKS;                                                                    02812000 T  0053
   LOOP;                                                                                 02813000 T  0055
   IF ADVANCE(NCR, ACR, CHRO, NCR, ACR) THEN                                             02814000 T  0055
      IF CONTINUE THEN                                                                   02815000 T  0057
%                                                                                        02816000 T  0058
         IF READACARD THEN                                                               02817000 T  0058
            BEGIN                                                                        02818000 T  0059
            IF LISTOG THEN PRINTCARD ;                                                   02818010 T  0059
            IF ACR.[33:15]≥ACCUMSTOP THEN                                                02818020 T  0061
               BEGIN XTA←BLANKS; FLOG(175); GO LOOPO END ;                               02818030 T  0062
            GO LOOP ;                                                                    02818040 T  0065
            END                                                                          02818050 T  0065
         ELSE IF T ← ACCUM[1] = "      " THEN GO TO CASE5 ELSE SCN ← 4                   02819000 T  0065
      ELSE IF T ← ACCUM[1] = "      " THEN                                               02820000 T  0068
            GO TO CASE3 ELSE SCN ← 3                                                     02821000 T  0072
   ELSE IF T ← ACCUM[1] = "      " THEN GO TO CASE2 ELSE SCN ← 2;                       02822000 T  0073
END;                                                                                     02823000 T  0078
CASE2;                                                                                   02824000 T  0079
BEGIN T ← CHAR; SCN ← 1 END;                                                             02825000 T  0080
CASE3;                                                                                   02826000 T  0082
BEGIN T ← "     "; NEXT ← SEMI; SCN ← 0;                                                 02827000 T  0083
      IF FOSTOG THEN IF LOGIFTOG THEN BEGIN LOGIFTOG ← FALSE; XTA ← T;                   02827100 T  0085
                  FLAG(101); END;                                                        02827200 T  0089
      GO TO XIT;                                                                         02827300 T  0090
END;                                                                                     02827400 T  0093
CASE4;                                                                          %994-    02827999 C  0093
BEGIN T ← "     "; NEXT ← SEMI; SCN ← 5; GO TO XIT END;                                  02828000 T  0094
CASE5;                                                                                   02829000 T  0098
BEGIN T ← "<EOF> "; NEXT ← EOF; EOSTOG ← FALSE; GO TO XIT END;                           02830000 T  0099
CASE6;                                                                          %994-    02830999 C  0103
BEGIN T ← ACCUM[1] ← NEXTACC; SCN ← NEXTSCN;                                             02831000 T  0104
   IF T = "      " THEN GO TO CASESTMT;                                                  02832000 T  0106
END;                                                                                     02833000 T  0107
CASE7;                                                                          %994-    02833999 C  0108
BEGIN EOSTOG ← TRUE;                                                                     02834000 T  0109
   IF LABELR THEN GO TO CASE5 ELSE GO TO CASE1;                                          02835000 T  0109
END;                                                                                     02836000 T  0111
CASE8;                                                                          %994-    02836999 C  0111
BEGIN T ← EXACCUM[1]; SCN ← 3 END;                                                       02837000 T  0112
CASE9;                                                                          %994-    02837999 C  0114
BEGIN T ← EXACCUM[1]; SCN ← 4 END;                                                       02838000 T  0115
CASE10;                                                                         %994-    02838999 C  0117
BEGIN T ← CHAR ← EXACCUM[0]; SCN ← 1  END;                                               02839000 T  0118
CASE11;                                                                         %994-    02839999 C  0121
BEGIN T ← EXACCUM[1]; SCN ← 10 END;                                                      02840000 T  0122
CASE12;                                                                         %994-    02840999 C  0124
BEGIN T ← EXACCUM[1]; SCN ← 1;                                                           02841000 T  0125
   IF N ← EXACCUM[2] ≤ 1 THEN                                                            02842000 T  0126
   BEGIN NEXT ← NUM; FNEXT ← N; GO TO XIT END;                                           02843000 T  0128
   NEXT ← 0;                                                                             02844000 T  0132
   OP ← N-1;                                                                             02845000 T  0132
   PREC ← IF N ≤ 4 THEN N-1 ELSE 4;                                                      02846000 T  0134
   GO TO XIT;                                                                            02847000 T  0137
END;                                                                                     02848000 T  0137
CASE13;                                                                         %994-    02848999 C  0138
BEGIN T ← "FUNCTI"; NEXT ← 16; SCN ← 6; GO TO XIT END;                                   02849000 T  0139
CASE14;                                                                         %994-    02849999 C  0143
```

```
BEGIN T ← ACCUM[1] ← NEXTACC;                                                                    02850000 T  0144
   NEXTACC ← NEXTACC2; SCN ← 6;                                                                  02851000 T  0145
END;                                                                                             02852000 T  0147
END OF CASE STATEMENT;                                                                           02853000 T  0147
                                                                                      START OF SEGMENT **********    89
                                                                                        89 IS   16 LONG, NEXT SEG    81
IF NOT FILETOG THEN                                                                              02854000 T  0148
   IF EOSTOG THEN                                                                                02855000 T  0148
      BEGIN                                                                                      02856000 T  0149
         NEXT ← 0;                                                                               02857000 T  0150
         IF T = "      " THEN GO TO CASESTMT;                                                    02858000 T  0150
         CHECKRESERVED;                                                                          02859000 T  0152
         IF NEXT > 0 THEN GO TO XIT;                                                             02860000 T  0152
      END;                                                                                       02861000 T  0153
IF (IDINFO←TIPE[T,[12:6]])>0 THEN                                                                02862000 T  0153
    BEGIN                                                                                        02862100 T  0155
BK: NEXT←ID ;                                                                                    02862200 T  0156
    IF NOT FILETOG THEN                                                                          02862300 T  0157
         IF SCANENTER←((FNEXT←SEARCH(T))=0) THEN FNEXT←ENTER(IDINFO,T)                           02862400 T  0158
         ELSE IF GET(FNEXT).CLASS=DUMMY THEN FNEXT←GET(FNEXT+2).BASE ;                           02862500 T  0162
    GO XIT ;                                                                                     02862600 T  0169
    END ;                                                                                        02862700 T  0170
GO CASEL[-IDINFO];    % SEE INITIALIZATION OF "TIPE", LINE 03433100%993-                         02862800 P  0170
L1:      %DIGITS                                                                      %993-      02863000 P  0172
BEGIN NUMTYPE ← INTYPE; NEXT ← NUM;  XTA ← ACRO;                                                  02864000 T  0173
    FNEXT ← DBLOW ← C1 ← 0;                                                                      02865000 T  0175
   XTA  ← CONVERT(FNEXT,C1,XTA ,TS);                                                             02866000 T  0177
   WHILE TS ≤ 9 DO                                                                               02867000 T  0179
      BEGIN                                                                                      02868000 T  0181
         XTA  ← CONVERT(F1,C1,XTA ,TS);                                                          02869000 T  0181
         IF (F2 ← FNEXT×TEN[C1]+F1) ≤ MAX                                                        02870000 T  0184
          THEN FNEXT ← F2                                                                        02871000 T  0186
           ELSE BEGIN                                                                            02872000 T  0187
                   NUMTYPE ← DOUBTYPE;                                                           02873000 T  0188
                   DOUBLE(FNEXT,DBLOW,TEN[C1],TEN[69+C1],X,                                      02874000 T  0189
                        F1,0,+,←,FNEXT,DBLOW);                                                   02875000 T  0191
                END;                                                                             02876000 T  0193
      END;                                                                                       02877000 T  0193
   IF CHAR = ".     " OR CHAR = "E     " OR CHAR = "D     " THEN                                02878000 T  0193
    CHECKPERIOD                                                                                  02879000 T  0197
   ELSE IF CHAR = "H     " THEN HOLLERITH;                                                       02883000 T  0197
   GO TO XIT;                                                                                    02884000 T  0204
END;                                                                                             02885000 T  0206
L2:      % >                                                                          %993-      02898100 P  0206
BEGIN PREC ← 4; OP ← 7; GO TO XIT END;                                                           02899000 T  0206
L3:      % ≥                                                                          %993-      02899100 P  0208
BEGIN PREC ← 4; OP ← 8; GO TO XIT END;                                                           02900000 T  0208
L4:      % & OR +                                                                     %993-      02900100 P  0210
BEGIN PREC ← 5; OP ← 10; NEXT ← PLUS; GO TO XIT END;                                             02901000 T  0210
L5:      % .                                                                          %993-      02910100 P  0212
BEGIN                                                                                            02911000 T  0213
    FNEXT ← DBLOW ← C1 ← 0; NUMTYPE ← REALTYPE;                                                  02912000 T  0213
   CHECKPERIOD;                                                                                  02913000 T  0215
      T ← FXACCUM[1];                                                                            02914000 T  0216
   IF SCN = 12 THEN                                                                              02915000 T  0217
   BEGIN SCN ← 1;                                                                                02916000 T  0217
   IF N ← FXACCUM[2] ≤ 1 THEN                                                                    02917000 T  0219
```

```
   BEGIN                                                                                    02918000 T  0220
      NEXT ← NUM; FNEXT ← N;                                                                02919000 T  0221
      NUMTYPE ← LOGTYPE; GO TO XIT;                                                         02920000 T  0222
   END;                                                                                     02921000 T  0223
   NEXT ← 0;                                                                                02922000 T  0223
   OP ← N=1;                                                                                02923000 T  0224
   PREC ← IF N ≤ 4 THEN N=1 ELSE 4;                                                         02924000 T  0225
   GO TO XIT;                                                                               02924100 T  0229
END;                                                                                        02925000 T  0229
   IF NEXT ≠ NUM THEN BEGIN NEXT ← NUM; XTA ← T; FLOG(141) END;                             02925100 T  0229
   GO TO XIT;                                                                               02926000 T  0233
END;                                                                                        02927000 T  0233
L6:      % % OR (                                                           %993-          02929100 P  0233
BEGIN NEXT ← LPAREN; GO TO XIT END;                                                         02930000 T  0234
L7:      % <                                                                %993-          02930100 P  0235
BEGIN PREC ← OP ← 4; GO TO XIT END;                                                         02931000 T  0236
L8:      % LETTER O                                                        %993-          02938100 P  0237
BEGIN IF DATATOG THEN IF CHECKOCTAL THEN GO TO XIT;                                         02939000 T  0238
    IDINFO←TIPE[12];  GO BK ;                                                               02939100 T  0240
END;                                                                                        02939200 T  0242
L9:      % $                                                              %993-          02942100 P  0242
BEGIN NEXT ← DOLLAR; GO TO XIT END;                                                         02943000 T  0242
L10:     % *                                                              %993-          02943100 P  0243
IF CHECKEXP(NCR, NCR, T) THEN                                                               02944000 T  0244
BEGIN PREC ← 9; OP ← 15; NEXT ← UPARROW; GO TO XIT END ELSE                                  02945000 T  0245
L11:                                                                                        02945100 T  0249
BEGIN PREC ← 7; OP ← 13; NEXT ← STAR; GO TO XIT END;                                        02946000 T  0250
L12:     % -                                                              %993-          02946100 P  0252
BEGIN PREC ← 5; OP ← 11; NEXT ← MINUS; GO TO XIT END;                                       02947000 T  0253
L13:     % ) OR [                                                         %993-          02947100 P  0255
BEGIN NEXT ← RPAREN; GO TO XIT END;                                                         02948000 T  0256
L14:     % ;                                                              %993-          02948100 P  0257
BEGIN NEXT ← SEMI; GO TO XIT END;                                                           02949000 T  0258
L15:     % ≤                                                              %993-          02949100 P  0259
BEGIN PREC ← 4; OP ← 5; GO TO XIT END;                                                      02950000 T  0260
L16:     % /                                                              %993-          02951100 P  0262
BEGIN PREC ← 7; OP ← 14; NEXT ← SLASH; GO TO XIT END;                                       02952000 T  0262
L17:     % ,                                                              %993-          02960100 P  0264
BEGIN NEXT ← COMMA; GO TO XIT END;                                                          02961000 T  0265
L18:     % ≠                                                              %993-          02962100 P  0266
BEGIN PREC ← 4; OP ← 9; GO TO XIT END;                                                      02963000 T  0267
L19:     % = OR ← OR #                                                    %993-          02963100 P  0269
BEGIN NEXT ← EQUAL; GO TO XIT END;                                                          02964000 T  0269
L20:     % ]                                                              %993-          02964100 P  0270
BEGIN XTA ← T; FLAG(0); GO TO CASESTMT END;                                                 02965000 T  0271
L21:     % " OR ' OR @                                                    %993-          02965100 P  0273
BEGIN QUOTESTRING; GO TO XIT END;                                                           02966000 T  0273
XIT:                                                                                        02971000 T  0274
IF DEBUGTOG THEN WRITALIST(FD,3,NEXT,T,"      ",0,0,0,0,0) ;                            02972000 T  0274
                                                                                            02973000 T  0278
   XTA ← NAME ← T;                                                                          02974000 T  0278
END SCAN;                                                                                   02975000 T  0279
                                                                                  81 IS  285 LONG, NEXT SEG    6
PROCEDURE WRAPUP;                                                                           02976000 T  0506
PRT(670) = WRAPUP
      COMMENT WRAPUP OF COMPILIATION;                                                       02977000 T  0506
    BEGIN                                                                                   02978000 T  0506
```

```
                ARRAY PRT[0:7,0:127],                                                    02979000 T  0506
                                                                                START OF SEGMENT **********    90
STACK(F+2) = PRT
                  SEGDICT[0:7,0:127],                                                    02980000 T  0002
STACK(F+3) = SEGDICT
                     SEG0[0:29];                                                         02981000 T  0004
STACK(F+4) = SEG0
                ARRAY FILES[0:BIGGESTFILENB];                                          02982000 T  0005
STACK(F+5) = FILES
                INTEGER THEBIGGEST;                                                    02983000 T  0007
STACK(F+6) = THEBIGGEST
                SAVE ARRAY FPB[0:1022]; % FILE PARAMETER BLOCK                         02984000 T  0007
STACK(F+7) = FPB
                REAL FPS,FPE;           % START AND END OF FPB                         02985000 T  0009
STACK(F+10) = FPS
STACK(F+11) = FPE
                REAL GSEG,PRI,FID,MFID,IDNM,FILTYP,FPBI;                               02986000 T  0009
STACK(F+12) = GSEG
STACK(F+13) = PRI
STACK(F+14) = FID
STACK(F+15) = MFID
STACK(F+16) = IDNM
STACK(F+17) = FILTYP
STACK(F+20) = FPBI
                BOOLEAN ALF;                                                           02987000 T  0009
STACK(F+21) = ALF
                REAL PRTADR, SEGMNT, LNK, TSEGSZ, T1, I, FPBSZ;                        02988000 T  0009
STACK(F+22) = PRTADR
STACK(F+23) = SEGMNT
STACK(F+24) = LNK
STACK(F+25) = TSEGSZ
STACK(F+26) = T1
STACK(F+27) = I
STACK(F+30) = FPBSZ
                    DEFINE                                                             02988900 T  0009
                        SPDEUN= FPBSZ#,                                                02988910 T  0009
                    ENDDEF=#;                                                          02988990 T  0009
                ARRAY INTLOC[0:150];                                                   02989000 T  0009
STACK(F+31) = INTLOC
                REAL J;                                                                02990000 T  0011
STACK(F+32) = J
                FORMAT SEGUS(A6, "  IS SEGMENT ", I4,                                 02991000 T  0011
                                                                                START OF SEGMENT **********    91
PRT(671) = SEGUS
                    ", PRT IS ", A4, ".");                                             02992000 T  0011
                                                                                 91 IS   12 LONG, NEXT SEG    90
                LIST SEGLS(IDNM,NXAVIL,T1);                                            02993000 T  0011
PRT(672) = *LIST, LABEL, OR SEGMENT DESCRIPTOR*
STACK(F+33) = SEGLS
                LABEL LA, ENDWRAPUP;                                                   02997000 T  0019
                 LABEL QQQDISKDEFAULT;                                          %503-  02997010 C  0019
                        COMMENT FORMAT OF SEGMENT DICTIONARY *RUN TIME ;                02998000 T  0019
                DEFINE      SGTYPF= [1:2]#, %0 = PROGRAM SEGMENTS                      02999000 T  0019
                            SGTYPC= 1:46:2#,%1 = MCP INTRINSIC                         03000000 T  0019
                                            %2 = DATA SEGMENT                          03001000 T  0019
                            PRTLINKF= [8:10]#,  % LINK TO FIRT PRT ENTRY               03002000 T  0019
                            PRTLINKC= 8:38:10#,                                        03003000 T  0019
```

```
                SGLCF   = [18:15]#, % SEGMENT SIZE                                  03004000 T  0019
                SGLCC   = 23:38:10#,                                                03005000 T  0019
                DKADRF  = [33:15]#, % RELATIVE DISK ADDRESS OF SEGMENT              03006000 T  0019
                                    % OR MCP INTRINSIC NUMBER                       03007000 T  0019
                DKADRC  = 33:13:15#;                                                03008000 T  0019
            COMMENT FORMAT OF FIRST SEGMENT OF CODE FILE- RUN TIME;                  03009000 T  0019
COMMENT SEG0[0:29]                                                                  03010000 T  0019
                WORD      CONTENTS                                                  03011000 T  0019
                 0        LOCATION OF SEGMENT DICTIONARY                            03012000 T  0019
                 1        SIZE OF SEGMENT DICTIONARY                                03013000 T  0019
                 2        LOCATION OF PRT                                           03014000 T  0019
                 3        SIZE OF PRT                                               03015000 T  0019
                 4        LOCATION OF FILE PARAMETER BLOCK                          03016000 T  0019
                 5        SIZE OF FILE PARAMETER BLOCK                              03017000 T  0019
                 6        STARTING SEGMENT NUMBER                                   03018000 T  0019
                 7-[2:1] IND FORTRAN FAULT DEC                                      03018100 T  0019
                 7-[18:15] NUMBER OF FILES                                          03019000 T  0019
                 7-[33:15] CORE REQUIRED/64                                         03020000 T  0019
                ;                                                                   03021000 T  0019
            COMMENT FORMAT OF PRT;                                                  03022000 T  0019
              %  FLGF = [0:4] = 1101 = SET BY STREAM                                03023000 T  0019
DEFINE          MODEF =[4:2]#,  % 0 = THUNK                                         03024000 T  0019
                MODEC=4:46:2#,  % 1 = WORD MODE PROGRAM DESCRIPTOR                  03025000 T  0019
                                % 2 = LABEL DESCRIPTOR                              03026000 T  0019
                                % 3 = CHARACTER MODE PROGRAM DESCRIPTOR             03027000 T  0019
                STOPF =[6:1]#,  % STOPPER = 1 FOR LAST DESCRIPTOR IN                 03028000 T  0019
                STOPC=6:47:1#,  % CHAIN OF SAME SEGMENT DESCRIPTORS                  03029000 T  0019
                LINKF =[7:11]#, % IF STOP = 0 THEN PRTLINK                         03030000 T  0019
                LINKC=7:37:11#, %              ELSE LINK TO SEGDICT                 03031000 T  0019
                FFF   =[18:15]#,% INDEX INTO SEGMENT DICTIONARY                     03032000 T  0019
                FFC   =18:33:15#,                                                   03033000 T  0019
                SINX = [33:15]#;% RELATIVE ADDRESS INTO SEGMENT                     03034000 T  0019
            DEFINE              PDR = [37:5]#,                                      03035000 T  0019
                                PDC = [42:6]#;                                      03036000 T  0019
            REAL STREAM PROCEDURE MKABS(F);                                         03037000 T  0019
PRT(673) = MKABS
                BEGIN                                                               03038000 T  0019
                    SI ← F; MKABS ← SI;                                             03039000 T  0020
                END MKABS;                                                          03040000 T  0020
            REAL STREAM PROCEDURE BUILDFPB(DEST,FILNUM,FILTYP,MFID,FID,IDSZ,        03041000 T  0021
PRT(674) = BUILDFPB
                                                IDNM,SPDEUN);                       03042000 T  0021
                VALUE     DEST,IDSZ,SPDEUN;                                         03043000 T  0021
                BEGIN                                                               03044000 T  0021
                    DI ← DEST;                                                      03045000 T  0022
                    SI ← FILNUM; SI ← SI + 6; DS ← 2 CHR;                           03046000 T  0022
                    SI ← FILTYP; SI ← SI + 7; DS ← CHR;                             03047000 T  0023
                    SI ← MFID; SI ← SI + 1; DS ← 7 CHR;                             03048000 T  0023
                    SI ← FID; SI ← SI + 1; DS ← 7 CHR;                              03049000 T  0024
                    SI ← LOC IDSZ; SI ← SI + 7; DS ← CHR;                           03050000 T  0025
                    SI ← IDNM; SI ← SI + 1; DS ← IDSZ CHR;                          03051000 T  0026
                    SI←LOC SPDEUN;SI←SI+6;DS←2 CHR;%   DISK SPEED & EU NUMBER+1     03051200 T  0027
                    BUILDFPB ← DI;                                                  03052000 T  0027
                    DS ← 2 LIT "0";                                                 03053000 T  0028
                END BUILDFPB;                                                       03054000 T  0028
            REAL STREAM PROCEDURE GITSZ(F);                                         03055000 T  0029
PRT(675) = GITSZ
```

```
                    BEGIN                                                                    03056000 T  0029
                         SI ← F; SI ←SI + 7; TALLY ← 7;                                      03057000 T  0030
                         3(IF SC ≠ " " THEN JUMP OUT;                                        03058000 T  0030
                           SI ←SI - 1; TALLY ← TALLY + 63;);                                 03059000 T  0032
                         GITSZ ← TALLY;                                                      03060000 T  0033
                    END GITSZ;                                                               03061000 T  0033
               STREAM PROCEDURE MOVE(F,T,SZ); VALUE SZ;                                      03062000 T  0034
PRT(676) = MOVE
                    BEGIN                                                                    03063000 T  0034
                         SI ← F; DI ←T; DS ← SZ WDS;                                         03064000 T  0035
                         END MOVE;                                                           03065000 T  0036
               INTEGER PROCEDURE MOVEANDBLOCK(FROM,SIZE); VALUE SIZE;                        03066000 T  0036
PRT(677) = MOVEANDBLOCK
                    ARRAY FROM[0,0]; INTEGER SIZE;                                           03067000 T  0036
                    BEGIN                                                                    03068000 T  0036
                         REAL T,NSEGS,J,I;                                                   03069000 T  0036
                                                                                  START OF SEGMENT **********   92
STACK(F+3) = T
STACK(F+4) = NSEGS
STACK(F+5) = J
STACK(F+6) = I
                    STREAM PROCEDURE M2(F,T); BEGIN SI←F; DI←T; DS ← 2 WDS; END M2;           03070000 T  0000
PRT(700) = M2
                         NSEGS ← (SIZE+29) DIV 30;                                           03071000 T  0001
                         IF DALOC DIV CHUNK < T ← (DALOC + NSEGS) DIV CHUNK                  03072000 T  0002
                           THEN DALOC ← CHUNK × T;                                           03073000 T  0004
                         MOVEANDBLOCK ← DALOC;                                               03074000 T  0007
                         DO BEGIN FOR J ← 0 STEP 2 WHILE J < 30 AND I<SIZE DO                03075000 T  0008
                          BEGIN                                                              03076000 T  0013
                              M2(FROM[I,[36:5],1,[41:7]],CODE(J));                           03077000 T  0013
                              I ← I+2;                                                       03078000 T  0018
                         END;                                                                03079000 T  0020
                              WRITE(CODE[DALOC]);                                            03080000 T  0020
                              DALOC ← DALOC +1;                                              03081000 T  0025
                            END UNTIL I ≥ SIZE;                                              03082000 T  0026
                    END MOVEANDBLOCK;                                                        03083000 T  0027
                                                                                  92 IS   32 LONG, NEXT SEG   90
               STREAM PROCEDURE MDESC(VLU,PRT); VALUE VLU;                                   03084000 T  0036
PRT(701) = MDESC
                    BEGIN                                                                    03085000 T  0036
                         DI ← LOC VLU; DS ← SFT;   % FLAG BIT                                03086000 T  0037
                         DI ← PRT; SI ← LOC VLU; DS ← WDS;                                   03087000 T  0037
                    END MDESC;                                                               03088000 T  0038
               INTEGER STREAM PROCEDURE MAKEINT(S);                                          03089000 T  0038
PRT(702) = MAKEINT
               BEGIN LABEL LOOP;     LOCAL T;                                              03090000 T  0038
               SI ← S;   SI ← SI + 3;                                                      03091000 T  0039
               LOOP: IF SC ≥ "0" THEN                                                      03092000 T  0039
                     BEGIN TALLY ← TALLY + 1;   SI ← SI + 1;  GO TO LOOP;   END;           03093000 T  0040
               T ← TALLY;   SI ← S;  SI ← SI + 3;  DI ← LOC MAKEINT;  DS ← T OCT;          03094000 T  0041
               END MAKEINT;                                                                03095000 T  0043
               LABEL FOUND;                                                                03096000 T  0044
               FORMAT                                                                      03097000 T  0044
                        FILEF(A1, A6, X1,                                                  03098000 T  0044
                                                                                  START OF SEGMENT **********   93
PRT(703) = FILEF
                        "FILE IDENTIFIER, PRT IS ", A4, ".");                              03099000 T  0044
```

```
     BLOKF(A6, X5, "COMMON BLOCK, PRT IS ", A4,                                  03100000 T  0044
          ", LENGTH IS ", I5, " WORDS,");                                        03101000 T  0044
                                                                          93 IS   27 LONG, NEXT SEG   90
     COMMENT DUMP OUT ACCUMULATED FORMAT AND NAME ARRAYS;                        03102000 T  0044
        IF FNNINDEX ≠ 0 THEN                                                     03103000 T  0044
        BEGIN FNNPRT ← PRGDESCBLDR(1,FNNPRT,0,NXAVIL ← NXAVIL + 1);              03104000 T  0045
              WRITEDATA(FNNINDEX,NXAVIL,FNNHOLD);                                03105000 T  0049
        END;                                                                     03106000 T  0051
         IF SAVESUBS > 0 THEN                                                    03106100 T  0051
           BEGIN T1 := GET(T := GLOBALSEARCH(",SUBAR")+2);                       03106125 T  0051
                 PUT(T,T1:=T1&SAVESUBS[TOSIZE]);                                 03106150 T  0055
           END;                                                                  03106175 T  0057
     T1←PRGDESCBLDR(1,23,0,NSEG←NXAVIL←NXAVIL+1) ; % BUILD TPAR AT               03106200 T  0057
     FILL LSTT[*] WITH 21(0),8("        ") ;       % R+23                       03106205 T  0060
                                                                        START OF SEGMENT *********   94
                                                                         94 IS   29 LONG, NEXT SEG   90
     WRITEDATA(29,NXAVIL,LSTT) ;                                                 03106210 T  0062
     PDPRT[(PD:NX-1),[37:5],(PD:NX-1),[42:6]],[6:1]←1 ; % SAVE BIT               03106215 T  0064
     T1 ← PRGDESCBLDR(1,22,0,NSEG ← NXAVIL ← NXAVIL + 1);                        03107000 T  0069
     WRITEDATA (138,NXAVIL,TEN);   % POWERS OF TEN TABLE                        03108000 T  0072
     IF  LSTI > 0 THEN                                                           03109000 T  0074
BEGIN                                                                            03110000 T  0075
     WRITEDATA(LSTI, NXAVIL ← NXAVIL+1, LSTP);                                   03111000 T  0075
     LSTA ← PRGDESCBLDR(1, LSTA, 0, NXAVIL);                                     03112000 T  0078
END;                                                                             03113000 T  0080
     IF TWODPRTX ≠ 0 THEN                                                        03114000 T  0080
BEGIN                                                                            03115000 T  0081
     FILL LSTT[*] WITH                                                           03116000 T  0081
     OCT0000000421410010,                                                        03117000 T  0082
                                                                        START OF SEGMENT *********   95
     OCT0301001301412025,                                                        03118000 T  0083
     OCT2021010442215055,                                                        03119000 T  0083
     OCT2245400320211025,                                                        03120000 T  0083
     OCT0106177404310415,                                                        03121000 T  0083
     OCT1025042112350000;                                                        03122000 T  0083
                                                                          95 IS    6 LONG, NEXT SEG   90
     T ← PRGDESCBLDR(0, TWODPRTX, 0, NXAVIL ← NXAVIL+1);                         03123000 T  0083
     WRITEDATA(-6, NXAVIL, LSTT);                                                03124000 T  0086
END;                                                                             03125000 T  0088
          COMMENT DECLARE GLOBAL FILES AND ARRAYS;                               03126000 T  0088
     FPS ← FPE ← MKABS(FPB);                                                     03127000 T  0088
     SEGMENTSTART;                                                               03128000 T  0091
     F2TOG ← TRUE;                                                               03129000 T  0091
     GSEG ← NSEG;                                                                03130000 T  0093
     FPBI ← 0;                                                                   03131000 T  0094
     EMITL(0); EMITL(2); EMITO(SSF);                                             03132000 T  0094
     EMITL(1);                 % SET BLOCK COUNTER TO 1                         03133000 T  0097
     EMITL(16); EMITO(STD);                                                      03134000 T  0097
     EMITL(0); EMITOPDCLIT(23); EMITO(DEL);                                      03138000 T  0099
     EMITL(REAL(HOLTOG)); EMITPAIR(21,STD);                                      03139000 T  0101
     I ← GLOBALNEXTINFO;  WHILE I < 4093 DO                                      03140000 T  0103
BEGIN                                                                            03141000 T  0106
     I ← I+3;                                                                    03142000 T  0106
     GETALL(I,INFA,INFB,INFC);                                                   03143000 T  0107
 IF INFA.CLASS = FILEID THEN   %SEE COMMENTS ON LINE 02118000  %992-             03144000 P  0109
BEGIN                                                                            03145000 T  0110
     FPBI ← FPBI + 1;                                                            03146000 T  0110
```

```
     PRI ← INFA    .ADDR;                                                                  03147000 T  0112
     IF (XTA ← INFB    ).[18:6] < 10 THEN                                                  03148000 T  0113
BEGIN                                                                                      03149000 T  0115
     IF XTA ← MAKEINT(XTA) > BIGGESTFILENB THEN FLAG(77) ELSE                               03150000 T  0115
     FILES[XTA] ← PRI;                                                                     03151000 T  0119
     IF XTA > THEBIGGEST THEN THEBIGGEST ← XTA;                                            03152000 T  0123
END;                                                                                       03153000 T  0125
     EMITO(MKS);                                                                           03154000 T  0125
     IF J ← INFC    .ADINFO ≠ 0 THEN    %  OPTION FILE                                    03155000 T  0126
BEGIN FILTYP ← INFC    .LINK;                                                              03156000 T  0127
     IDNM ← "      "&"FILE"[6:24:24]&INFB[30:18:18];                                     03157000 T  0129
     T1 ← GITSZ(IDNM);                                                                     03158000 T  0132
     FID ← FILEINFO[2,J];                                                                  03159000 T  0134
     MFID ← FILEINFO[1,J];                                                                 03160000 T  0135
     IF FILTYP≥10 AND (T←FILEINFO[3,J].DKAREASZ)≠0 THEN                                    03161000 T  0137
          BEGIN %%% SET UP <DISK FILE DESCRIPTION>;                                        03162000 T  0141
          SPDEUN:=FILEINFO[3,J].SENSPDEUNF;                                                03162005 T  0141
          B←IF (B←((J←FILEINFO[0,J]).[18:12])/(IF A←J.[30:12]≤0 THEN                       03162007 T  0144
            1 ELSE A))≤0 THEN 1 ELSE B ;                                                   03162014 T  0147
               %%% B=ORIGINAL "BLOCKING" SIZE = # LOGRECS/PHYSREC.                         03162020 T  0152
          A←ENTIER(B×ENTIER(T/(20×B)+.999999999)+.5) ;                                     03162030 T  0152
               %%% T="AREA" SIZE = # LOGRECS IN TOTAL FILE.                               03162040 T  0158
               %%% A=# LOGRECS PER ROW.                                                    03162050 T  0158
          B←ENTIER(T/A+.999999999) ;                                                       03162060 T  0158
               %%% B = # ROWS IN FILE.                                                    03162070 T  0160
               %%% EQUIVALENT ALGOL FILE DESCRIPTION = [B:A].                              03162080 T  0160
               %%% THE ABOVE LOGIC YIELDS: SHORTEST ROW CONTAINING                         03162090 T  0160
               %%% AN INTEGER NUMBER OF PHYSICAL RECORDS AND WHICH                         03162100 T  0160
               %%% REQUIRES 20 OR FEWER ROWS FOR THE TOTAL AREA. T.                        03162110 T  0160
          EMITNUM(B); EMITNUM(A) ;                                                         03162120 T  0160
          END ELSE                                                                         03162129 T  0162
          BEGIN EMITL(0); EMITL(0);                                                        03163000 T  0162
               J ← FILEINFO[0,J]; % THIS ONE HAS ALL THE GOODIES                           03164000 T  0168
          END;                                                                             03164010 T  0170
QQQDISKDEFAULT;                                                          %503-             03164045 C  0170
          ESTIMATE←ESTIMATE+(J.[42:6])×(IF A←J.[18:12]=0 THEN J.[30:12]                    03164050 T  0171
               ELSE A) ;                                                                   03164100 T  0174
          EMITL(J.[4:2]);    %  LOCK                                                      03165000 T  0176
          EMITL(FPBI);    %   FILE PARAM INDEX                                           03166000 T  0178
                                                                                         03167000 T  0178
          EMITDESCLIT(PRI);    %  PRT OF FILE                                            03168000 T  0178
          EMITL(J.[42:6]);    %  # BUFFERS                                               03169000 T  0179
          EMITL(J.[3:1]);  %  RECORDING MODE                                            03170000 T  0180
          EMITNUM(J.[30:12]) ;  % RECORD SIZE                                          03171000 T  0182
          EMITNUM(J.[18:12]) ;  % BLOCK SIZE                                          03172000 T  0183
          EMITNUM(J.[ 6:12]) ;  % SAVE FACTOR                                        03173000 T  0184
     END ELSE                                                                          03174000 T  0185
     BEGIN                                                                             03175000 T  0185
          ALF ←TRUE;                                                                   03176000 T  0186
          IF(FILTYP←INFC.LINK=2 OR FILTYP=12)AND INFB.[18:6]≤9 THEN                    03177000 T  0187
            IDNM ← 0&"FILE"[6:24:24]&INFB[30:18:18]                                    03178000 T  0191
               ELSE                                                                    03179000 T  0194
     BEGIN                                                                             03180000 T  0194
          ALF ← FALSE;                                                                 03181000 T  0196
               IF (IDNM ← "    "&INFB[6:18:30]) = "READR  " THEN                       03182000 T  0196
                    IDNM ← "READER ";                                                  03183000 T  0199
     END;                                                                              03184000 T  0200
```

```
IF IDNM="READER " OR IDNM="FILE5  " THEN IDNM←"CARD   " ELSE      %503-      03184010 C  0200
IF IDNM="FILE6  " THEN BEGIN IDNM←"PRINTER";FILTYP←18;END ELSE    %503-      03184020 C  0203
 BEGIN                                                            %503-      03184030 C  0211
 EMITL(20); EMITL(600); FILTYP←12; %20 x 600 REC DISK             %503-      03184040 C  0215
    J←0&2[42:42:6]&10[30:36:12]&300[18:36:12];                    %503-      03184050 C  0217
 FID←IDNM; MFID←"FORTEMP"; T1←GITSZ(IDNM);                        %503-      03184054 C  0221
     GO TO QQQDISKDEFAULT;                                        %503-      03184060 C  0224
 END;                                                             %503-      03184070 C  0226
        T1 ← GITSZ(IDNM);                                                    03185000 T  0226
        FID ← IDNM;                                                          03186000 T  0227
        MFID ← 0;                                                            03187000 T  0228
     IF DCINPUT AND ALF THEN BEGIN                                           03187100 T  0229
        EMITL(20);            % DISK ROWS                                    03187200 T  0231
        EMITL(100);           % DISK RECORD PER ROW                          03187300 T  0231
        EMITL(2);             % REWIND AND LOCK                              03187400 T  0232
        EMITL(FPBI);          % FILE NUMBER                                  03187450 T  0233
        EMITDESCLIT(PRI);     % PRT OF FILE                                  03187500 T  0234
        EMITL(2);             % NUMBER OF BUFFERS                            03187550 T  0234
        EMITL(1);             % RECORDING MODE                               03187600 T  0235
        EMITL(10);            % RECORD SIZE                                  03187650 T  0236
        EMITL(30);            % BLOCK SIZE                                   03187700 T  0237
        EMITL(1);             % SAVE FACTOR                                  03187750 T  0237
     END ELSE                                                                03187800 T  0238
     BEGIN                                                                   03187900 T  0238
        EMITL(0);       % DISK ROWS                                          03188000 T  0239
        EMITL(0);       % DISK RECORDS PER ROW                               03189000 T  0239
        EMITL(0);       % REWIND & RELEASE                                   03190000 T  0240
        EMITL(FPBI);    % FILE NUMBER                                        03191000 T  0241
                                                                             03192000 T  0242
        EMITDESCLIT(PRI);  % PRT OF FILE                                     03193000 T  0242
     EMITL(2);          % 2 BUFFERS                                          03194000 T  0242
     EMITL(REAL(ALF));                                                       03195000 T  0243
     EMITL(IF FILTYP = 0 THEN 10 ELSE 17);                                   03196000 T  0244
     EMITL(0);       % 15 WORD BUFFERS                                       03197000 T  0247
     EMITL(0);          % SAVE FACTOR (SCRATCH BY DEFAULT)                   03198000 T  0247
  END;                                                                       03198500 T  0248
END;                                                                         03199000 T  0248
     EMITL(11);         % INPUT OR OUTPUT                                   03200000 T  0248
     EMITL(8);          % SWITCH CODE FOR BLOCK                             03201000 T  0249
     EMITOPDCLIT(5);    % CALL BLOCK                                        03202000 T  0250
     FPE←BUILDFPB(FPE,FPBI,FILTYP,MFID,FID,T1,IDNM,SPDEUN);                  03203000 T  0250
     IF PRTOG THEN WRITALIST(FILEF,3,IDNM,[6:6],IDNM,B2D(PRI),               03204000 T  0254
     0,0,0,0,0) ;                                                            03204010 T  0258
END                                                                          03205000 T  0260
     ELSE                                                                    03206000 T  0260
     IF INFA.CLASS = BLOCKID THEN                                            03207000 T  0260
BEGIN                                                                        03208000 T  0261
     IF PRTOG THEN WRITALIST(BLOKF,3,INFB,B2D(INFA.ADDR),                    03209000 T  0262
     INFC.SIZE,0,0,0,0,0) ;                                                  03210000 T  0266
     IF INFA < 0 THEN ARRAYDEC(I);                                           03211000 T  0268
END;                                                                         03212000 T  0270
     IF (T1 ← INFA   .CLASS) ≥ FUNID                                         03213000 T  0270
      AND T1 ≤ SUBRID THEN                                                   03214000 T  0271
BEGIN      % CHECK FOR INTRINSIC                                             03215000 T  0273
     PRI ← 0;                                                                03216000 T  0273
     IF INFA   .SEGNO = 0 THEN                                               03217000 T  0274
BEGIN                                                                        03218000 T  0275
```

```
      A←0;  B←NUMINTM1 ;                                                                  03219000 T  0276
      WHILE A+1 < B DO                                                                    03220000 T  0277
      BEGIN                                                                               03221000 T  0279
         PRI ← REAL(BOOLEAN(A+B) AND BOOLEAN(1022));                                      03222000 T  0279
         IF IDNM ← INT[PRI] = INFB THEN GO TO FOUND;                                      03223000 T  0281
         IF INFB < IDNM THEN B ← PRI.[36:11] ELSE A ← PRI.[36:11];                        03224000 T  0283
      END;                                                                                03225000 T  0287
      IF IDNM ← INT[PRI←(A+B)×2-PRI] = INFB THEN GO TO FOUND;                             03226000 T  0288
      XTA ← INFB; FLAG(30);                                                               03227000 T  0292
      GO TO LA;                                                                           03228000 T  0293
FOUND;                                                                                    03229000 T  0294
      IF (T1←INT[PRI+1].INTPARMS)≠0                                                       03230000 T  0295
      AND INFC    < 0                                                                     03231000 T  0297
      THEN IF T1 ≠ INFC.NEXTRA THEN                                                       03232000 T  0297
         BEGIN XTA ← INFB    ; FLAG(28); END;                                             03233000 T  0300
      IF (FID←INTLOC[MFID←INT[PRI+1].INTNUM])=0 THEN                                      03234000 T  0302
BEGIN                                                                                     03235000 T  0305
      PDPRT[PDIR,PDIC] ←                                                                  03236000 T  0306
        0&1[STYPC]                                                                        03237000 T  0308
         &MFID[DKAC]                                                                      03238000 T  0309
         &(FID ← INTLOC[MFID] ← NXAVIL ← NXAVIL + 1)[SGNOC]                               03239000 T  0310
         &1[SEGSZC];                                                                      03240000 T  0314
      PDINX ← PDINX + 1;                                                                  03241000 T  0315
END;                                                                                      03242000 T  0316
      T1 ← PRGDESCBLDR(1,INFA    .ADDR,0,FID);                                            03243000 T  0316
      IF PRTOG THEN WRITALIST(SEGUS,3,IDNM,FID,B2D(T1),0,0,0,0,0) ;                       03244000 T  0319
      IF INT[PRI+1] < 0 THEN                                                              03245000 T  0324
BEGIN                                                                                     03246000 T  0326
      T1 ← PRGDESCBLDR(1,INT[PRI+1].INTPRT,0,FID);                                        03247000 T  0326
      INT[PRI+1] ← ABS(INT[PRI + 1]);                                                     03248000 T  0329
      END;                                                                                03249000 T  0332
      END                                                                                 03250000 T  0332
           ELSE IF PRTOG THEN WRITALIST(SEGUS,3,INFB,                                     03251000 T  0332
           INFA.SEGNO,B2D(INFA.ADDR),0,0,0,0,0) ;                                         03252000 T  0335
      END;                                                                                03253000 T  0339
      LA:                                                                                 03254000 T  0339
      END;                                                                                03255000 T  0340
COMMENT MUST FOLLOW THE FOR STATEMENT;                                                    03256000 T  0340
IF FILEARRAYPRT ≠ 0 THEN                                                                  03257000 T  0340
BEGIN    %  BUILDING OBJECT TIME FILE SEARCH ARRAY                                       03258000 T  0341
        J ← PRGDESCBLDR(1,FILEARRAYPRT,0,NXAVIL ← NXAVIL + 1);                            03259000 T  0341
        WRITEDATA(THEBIGGEST + 1,NXAVIL,FILES);                                           03260000 T  0344
END;                                                                                      03261000 T  0347
           XTA ← BLANKS;                                                                  03262000 T  0347
           IF NXAVIL > 1023 THEN FLAG(45);                                                03263000 T  0347
           IF PRTS > 1023 THEN FLAG(46);                                                  03264000 T  0349
           IF STRTSEG = 0 THEN FLAG(65);                                                  03265000 T  0351
           PRI ← 0;                                                                       03266000 T  0353
           WHILE (IDNM ← INT[PRI]) ≠ 0 DO                                                 03267000 T  0354
           IF INT[PRI+1] ≥ 0 THEN PRI ← PRI + 2 ELSE                                      03268000 T  0357
      BEGIN                                                                               03269000 T  0360
           IF (FID←INTLOC[MFID←INT[PRI+1].INTNUM])=0 THEN                                 03270000 T  0360
      BEGIN                                                                               03271000 T  0364
           PDPRT[PDIR,PDIC] ←                                                             03272000 T  0364
             0&1[STYPC]                                                                   03273000 T  0366
              &MFID[DKAC]                                                                 03274000 T  0368
              &(FID ← INTLOC[MFID] ← NXAVIL ← NXAVIL + 1)[SGNOC]                          03275000 T  0369
```

```
        &1[SEGSZC];
    PDINX ← PDINX + 1;
END;
    T1 ← PRGDESCBLDR(1,INT[PRI + 1],INTPRT,0,FID);
    PRI ← PRI+2;
END;
    FOR I ← 1 STEP 1 UNTIL BDX DO
    BEGIN EMITO(MKS); EMITOPDCLIT(BDPRT[I]) END;
    EMITO(MKS);
    EMITOPDCLIT(STRTSEG,[18:15]);
    T ← PRGDESCBLDR(1,0,0,NSEG);
    SEGMENT((ADR+4) DIV 4,NSEG,FALSE,EDOC);
    IF ERRORCT ≠ 0 THEN GO TO ENDWRAPUP;
    FILL SEGO[*] WITH
         0CT020005,      % BLOCK

         0CT220014,      % WRITE
         0CT230015,      % READ
         0CT240016;      % FILE CONTROL

         COMMENT INTRINSIC FUNCTIONS;
    FOR I ← 0 STEP 1 UNTIL 3 DO
BEGIN
    T1 ← PRGDESCBLDR(1,SEGO[I],[36:12],0,
                     NSEG ← NXAVIL ← NXAVIL +1);
    PDPRT[PDIR,PDIC] ←
      0&1[STYPC]
       &(SEGO[I],[30:6])[DKAC]
       &NXAVIL[SGNOC]
       &1[SEGSZC];
    PDINX ← PDINX + 1;
END;
         COMMENT GENERATE PRT AND SEGMENT DICTIONARY;
    PRT[0,41] ← PDPRT[0,0] & 63[10:42:6];  % USED FOR FAULT OPTN
    FOR I ← 1 STEP 1 UNTIL PDINX-1 DO
    IF (T1←PDPRT[I,PDR,I,PDC]).SEGSZF = 0 THEN
BEGIN           % PRT ENTRY
    PRTADR ←T1,PRTAF;
    SEGMNT ←T1,SGNOF;
    LNK ← SEGDICT[SEGMNT,[36:5], SEGMNT,[41:7]],PRTAF;
    MDESC(T1,RELADF&SEGMNT[FFC]
        &(REAL(LNK=0))[STOPC]
        &(IF LNK = 0 THEN SEGMNT ELSE LNK)[LINKC]
         &(T1.DTYPF)[MODEC]
         &5[1:45:3],
         PRT[PRTADR,[36:5],PRTADR,[41:7]]);
    SEGDICT[SEGMNT,[36:5],SEGMNT,[41:7]],PRTLINKF ← PRTADR;
END
    ELSE
BEGIN           % SEGMENT DICTIONARY ENTRY
    SEGMNT ← T1,SGNOF;
    SEGDICT[SEGMNT,[36:5],SEGMNT,[41:7]]←
    SEGDICT[SEGMNT,[36:5],SEGMNT,[41:7]]
        &T1[SGLCC]
        &T1[DKADRC]
        & T1[4:12:1]
        &T1[6:6:1]
```

```
                03276000 T  0372
                03277000 T  0374
                03278000 T  0375
                03279000 T  0375
                03280000 T  0378
                03281000 T  0379
                03282000 T  0380
                03283000 T  0381
                03284000 T  0385
                03285000 T  0385
                03288000 T  0387
                03289000 T  0389
                03290000 T  0392
                03291000 T  0393
                03292000 T  0394
START OF SEGMENT **********    96
                03293000 T  0395
                03294000 T  0395
                03295000 T  0395
 96 IS    4 LONG, NEXT SEG    90
                03296000 T  0395
                03297000 T  0395
                03298000 T  0396
                03299000 T  0396
                03300000 T  0397
                03301000 T  0400
                03302000 T  0402
                03303000 T  0403
                03304000 T  0405
                03305000 T  0406
                03306000 T  0408
                03307000 T  0409
                03308000 T  0411
                03308100 T  0411
                03309000 T  0415
                03310000 T  0419
                03311000 T  0423
                03312000 T  0424
                03313000 T  0425
                03314000 T  0426
                03315000 T  0429
                03316000 T  0431
                03317000 T  0432
                03318000 T  0435
                03319000 T  0437
                03320000 T  0438
                03321000 T  0440
                03322000 T  0445
                03323000 T  0445
                03324000 T  0445
                03325000 T  0445
                03326000 T  0446
                03327000 T  0449
                03328000 T  0451
                03329000 T  0452
                03329100 T  0453
                03329200 T  0454
```

```
        &T1[1:1:2];                                                          03330000 T  0455
    TSEGSZ ← TSEGSZ + T1.SEGSZF;                                             03331000 T  0456
END;                                                                         03332000 T  0458
      COMMENT WRITE OUT FILE PARAMETER BLOCK;                                03333000 T  0459
    FPBSZ ← ((FPE.[33:15] - FPS) × 8 + FPE.[30:3] + 9) DIV 8;                03334000 T  0459
    I ← (FPBSZ + 29) DIV 30;                                                 03335000 T  0463
    IF DALOC DIV CHUNK < T1 ← (DALOC +I) DIV CHUNK                           03336000 T  0465
      THEN DALOC ← CHUNK × T1;                                               03337000 T  0466
    SEGO[4] ← DALOC;                                                         03338000 T  0469
    SEGO[5] ← FPBSZ;                                                         03339000 T  0470
    SEGO[5].FPBVERSF←FPBVERSION;                                             03339100 T  0472
    FOR I ← 0 STEP 30 WHILE I < FPBSZ DO                                     03340000 T  0474
BEGIN                                                                        03341000 T  0478
                                                                             03342000 T  0478
    MOVE(FPB[I],CODE(0),IF (FPBSZ-I) ≥ 30                                    03343000 T  0478
                        THEN 30 ELSE (FPBSZ-I));                             03344000 T  0483
    WRITE(CODE[DALOC]);                                                      03345000 T  0485
    DALOC ← DALOC + 1;                                                       03346000 T  0490
END;                                                                         03347000 T  0491
    SEGO[2] ← MOVEANDBLOCK(PRT,PRTS+1); % WRITES OUT PRT                     03348000 T  0492
                                        % SAVES ADDRESS OF PRT               03349000 T  0495
    SEGO[3] ← PRTS + 1;      % SIZE OF PRT                                 03350000 T  0495
    SEGO[0] ← MOVEANDBLOCK(SEGDICT,NXAVIL + 1); % WRITE SEG DICT            03351000 T  0497
    SEGO[1] ← NXAVIL + 1;          % SIZE OF SEGMENT DICTIONARY             03352000 T  0500
    SEGO[6] ← -GSEG;   % FIRST SEGMENT TO EXECUTE                           03353000 T  0502
    SEGO[7].[33:15] ← FPBI;   % NUMBER OF FILES                            03354000 T  0503
    SEGO[7].[18:15] ← ESTIMATE ← IF    % CORE ESTIMATE                      03355000 T  0505
                        ( I ←                                                03356000 T  0506
                        ESTIMATE+60+ %%% OPTION FILE BUFF SIZES + DEFAULT BUFF SIZES,  03356100 T  0506
                        PRTS + 512   % PRT AND STACK SIZE                    03357000 T  0507
                       +TSEGSZ       % TOTAL SIZE OF CODE                   03358000 T  0508
                       + 1022   % FOR INTRINSICS                            03358050 T  0508
                       +ARYSZ        % TOTAL ARRAY SIZE                     03359000 T  0509
                       + (MAXFILES × 28) % SIZE OF ALL FIBS                  03360000 T  0509
                       +FPBSZ        % SIZE OF FILE PARAMETER BLOCK          03361000 T  0510
                       + (IF ESTIMATE = 0 THEN 0 ELSE (ESTIMATE + 1000))     03361100 T  0511
                       + (NXAVIL + 1)     % SIZE OF SEGMENT DICTIONARY       03362000 T  0514
                       ) > 32768 THEN 510 ELSE (I DIV 64);                   03363000 T  0515
                     COMMENT IF SEGSW THEN UPDATE LINDICT, SEGO[0] & WRITE IT ;  03363100 T  0519
                     SEGO[7].[2:1] ← 1;  % USED FOR FORTRAN FAULT DEC;       03363150 T  0519
                   IF SEGSW THEN                                             03363200 T  0522
                   BEGIN                                                     03363300 T  0523
                     FOR I ← NXAVIL + 1 STEP -1 UNTIL 1 DO                   03363400 T  0523
                      IF LINEDICT[I.IR,I.IC] = 0 THEN % INDICATE NO LINE SEGMENT  03363500 T  0527
                         LINEDICT[I.IR,I.IC] ← -1;    % FOR THIS SEGMENT    03363600 T  0530
                     SEGO[0] ← SEGO[0] & (MOVEANDBLOCK(LINEDICT,NXAVIL+1))[TOBASE];  03363700 T  0536
                   END;                                                      03363800 T  0540
                     WRITE(CODE[0],30,SEGO[*]);                              03364000 T  0540
                     IF ERRORCT = 0 AND SAVETIME ≥ 0 THEN LOCK(CODE);        03365000 T  0545
                ENDWRAPUP;                                                   03366000 T  0549
                     LOCK(TAPE);        %RW/L TAPE FILE OR LOCK DISK        %502-  03366100 P  0549
                     IF NTAPTOG THEN LOCK(NEWTAPE,*); %RW/L TAPE OR CRUNCH DISK%502-  03366200 C  0550
                END WRAPUP;                                                  03367000 T  0553
                                                                  90 IS  564 LONG, NEXT SEG    6
          PROCEDURE INITIALIZATION;                                          03368000 T  0506
PRT(704) = INITIALIZATION                                                    03368100 T  0506
          BEGIN COMMENT   INITIALIZATION;                                    03369000 T  0506
```

```
                ALPHA STREAM PROCEDURE MKABS(P);                                                    03370000 T  0506
                                                                                          START OF SEGMENT **********   97
PRT(705) = MKABS
                BEGIN S1 = P; MKABS = SI END;                                               03371000 T  0000
                STREAM PROCEDURE BLANKOUT(CRD, N); VALUE N;                                 03372000 T  0001
PRT(706) = BLANKOUT
                BEGIN D1 = CRD; N(DS = LIT " ") END;                                        03373000 T  0001
                BLANKOUT(CRD[10], 40);                                                      03374000 T  0004
                BLANKOUT(LASTSEQ, 8);                                                       03374100 T  0006
                BLANKOUT(LASTERR,8) ;                                                       03374200 T  0007
                INITIALNCR = MKABS(CRD[0])&6[30:45:3];                                      03375000 T  0008
                CHR0       = MKABS(ACCUM[0])& 2[30:45:3];                                   03376000 T  0011
                ACR0 = CHR0+1;                                                              03377000 T  0014
                ACR1 = (CHR1+MKABS(EXACCUM[0]) & 2[30:45:3]) +1;                            03378000 T  0016
                ACCUMSTOP=MKABS(ACCUM[11]);  EXACCUMSTOP=MKABS(EXACCUM[11]) ;               03378100 T  0020
                BUFL = MKABS(BUFF) & 2[30:45:3];                                            03379000 T  0024
                NEXTCARD = 1;                                                               03380000 T  0026
                GLOBALNEXTINFO = 4093;                                                      03381000 T  0027
                PDINX = 1;                                                                  03381100 T  0028
                LASTNEXT=1000 ;                                                             03381200 T  0029
                PRTS  = 41; % CURRENTLY . . . . . LAST USED PRT                             03382000 T  0029
                READ(CR, 10, CB[*]);                                                        03383000 T  0030
                LISTOG=TRUE;  SINGLETOG=TRUE; CHECKTOG = FALSE;  %DEFAULT          %501-    03384000 P  0034
                FIRSTCALL = TRUE;                                                           03385000 T  0040
                IF BOOLFAN(ERRORCT,[46:1]) THEN LISTOG = FALSE;                             03385100 T  0041
                IF BOOLFAN(ERRORCT,[47:1]) THEN DCINPUT = TRUE;                             03385200 T  0044
                ERRORCT = 0;                                                                03385300 T  0047
                IF DCINPUT THEN SEGSW = TRUE ;                                              03385350 T  0048
                IF DCINPUT THEN REMOTETOG=TRUE ;                                            03385355 T  0051
                LIMIT=IF DCINPUT THEN 20 ELSE 100 ;                                         03385360 T  0054
                IF SEGSW THEN SEGSWFIXED = TRUE ;                                           03385400 T  0057
                EXTRAINFO[0,0] = 0 & EXPCLASS[TOCLASS];                                     03386000 T  0060
                NEXTEXTRA = 1;                                                              03387000 T  0063
                LASTMODF = 1 ;                                                              03387100 T  0064
                DALOC = 1;                                                                  03388000 T  0064
                TYPE = -1;                                                                  03389000 T  0065
                   MAP[0] = MAP[2] = MAP[4] = MAP[7] = -10;                                 03390000 T  0066
                   MAP[5] = 1; MAP[6] = 2;                                                  03391000 T  0071
                FILL XR[*] WITH 0,0,0,0,0,0,0,                                              03391100 T  0073
                                                                                          START OF SEGMENT **********   98
                    "INTEGE","R    R","      ","      "," REAL ","      ",                 03391200 T  0075
                    "LOGICA","L    L","DOUBLE","      ","COMPLE","X    X",               03391300 T  0075
                    "------","=    =","      ","      "," ---- ","      ",               03391400 T  0075
                    "------","=    =","------","      ","------","=    =";              03391450 T  0075
                                                                                            98 IS   31 LONG, NEXT SEG   97
                FILL TYPES[*] WITH "      ", "INTGER", "      ", "REAL  ",                  03392000 T  0075
                                                                                          START OF SEGMENT **********   99
                   "LOGCAL", "DOUBLE", "COMPLX";                                            03393000 T  0077
                                                                                            99 IS    7 LONG, NEXT SEG   97
                FILL KLASS[*] WITH                                                          03394000 T  0077
                   "NULL  ", "ARRAY ", "VARBLE", "STFUN ",                                  03395000 T  0077
                                                                                          START OF SEGMENT ********** 100
                   "NAMLST", "FORMAT", "ERROR ", "FUNCTN",                                  03396000 T  0078
                   "INTRSC", "EXTRNL", "SUBRTN", "COMBLK",                                  03397000 T  0078
                   "FILE  ";                                                                03398000 T  0078
                                                                                           100 IS   13 LONG, NEXT SEG   97
                FILL RESERVEDWORDSLP[*] WITH                                                03399000 T  0078
```

```
    "CALL  ","ENTR  ","FORM  ","GOTO  ","IF    ","READ  ",

    "REAL  ","WRIT  ","DATA  ","CLOS  ","LOCK  ","PURG  ","CHAI  ",
    "PRIN  ","PUNC  ",
    0,"Y       ","AT      ",0,0,0,0,"E       ",0,"E       ",0,"E       ",
    "N       ","T       ","H       ";

FILL RESERVEDWORDS[*] WITH
    "ASSI  ","BACK  ","BLOC  ","CALL  ","COMM  ","COMP  ","CONT  ",

    "DATA  ","DIME  ","DOUB  ","END   ","ENDF  ","ENTR  ","EQUI  ",
    "EXTE  ","FUNC  ","GOTO  ","INTE  ","LOGI  ","NAME  ","PAUS  ",
    "PRIN  ","PROG  ","PUNC  ","READ  ","REAL  ","RETU  ","REWI  ",
    "STOP  ","SUBR  ","WRIT  ",
    "CLOS  ","LOCK  ","PURG  ",
    0,0,0,
    "FIXE  ","VARY  ","AUXM  ","RELE  ",
    "IMPL  ",
    "GN      ","SPACE   ","KDATA   ",0,"ON      ","LEX     ","INUE    ",
    0,"NSION   ","LEPRECIS",0,"ILE     ","Y       ","VALENCE ","RNAL    "
    ,"TION    ",0,"GER     ","CAL     ","LIST    ","E       ","T       ",
    "RAM     ","H       ",0,0,"RN      ","ND      ",0,"OUTINE  ",
    "E       ","E       ",0,"E       ",0,0,0,"D       ","ING     ",
    "EM      ","ASE     "
   ,"ICIT    "
    ;

FILL RESLENGTHLP[*] WITH
    4,5,6,4,2,4,4,5,4,5,4,5,5,5,5;

FILL LPGLOBAL[*] WITH
     4, 13, 36, 17, 35, 25,

    26, 31,  8, 32, 33, 34, 37, 22, 24;

FILL RESLENGTH[*] WITH
    0, 9, 9, 4, 6,

    7, 8, 4, 9, 15,
    3, 7, 5, 11, 8,
    8, 4, 7, 7, 8,
    5, 5, 7, 5, 4,
    4, 6, 6, 4, 10, 5,
    5, 4, 5, 0, 0, 0, 5, 7, 6, 7
   ,8
    ;

        FILL WOP[*] WITH
        "LITC","    ",

        "OPDC","DESC",
  10,"DFL ", 11,"NOP ", 12,"XRT ", 16,"ADD ", 17,"AD2 ", 18,"PRL ",
  19,"LNG ", 21,"GEQ ", 22,"BBC ", 24,"INX ", 35,"LOR ", 37,"GTR ",
  38,"BFC ", 39,"RTN ", 40,"COC ", 48,"SUB ", 49,"SB2 ", 64,"MUL ",
  65,"ML2 ", 67,"LND ", 68,"STD ", 69,"NEQ ", 70,"SSN ", 71,"XIT ",
  72,"MKS ",
```

```
                03400000 T  0079
          START OF SEGMENT **********  101
                03401000 T  0080
                03401100 T  0080
                03401200 T  0080
                03401300 T  0080
           101 IS   30 LONG, NEXT SEG   97
                03402000 T  0080
                03403000 T  0081
          START OF SEGMENT **********  102
                03404000 T  0082
                03405000 T  0082
                03406000 T  0082
                03407000 T  0082
                03407100 T  0082
                03407101 T  0082
                03407102 T  0082
                03407103 T  0082
                03407200 T  0082
                03407300 T  0082
                03407400 T  0082
                03407500 T  0082
                03407600 T  0082
                03407601 T  0082
                03407602 T  0082
                03407990 T  0082
           102 IS   84 LONG, NEXT SEG   97
                03408000 T  0082
                03409000 T  0082
          START OF SEGMENT **********  103
           103 IS   15 LONG, NEXT SEG   97
                03410000 T  0084
                03411000 T  0084
          START OF SEGMENT **********  104
                03411100 T  0085
           104 IS   15 LONG, NEXT SEG   97
                03412000 T  0085
                03413000 T  0086
          START OF SEGMENT **********  105
                03414000 T  0087
                03415000 T  0087
                03416000 T  0087
                03417000 T  0087
                03418000 T  0087
                03418100 T  0087
                03418101 T  0087
                03418990 T  0087
           105 IS   42 LONG, NEXT SEG   97
                03419000 T  0087
                03420000 T  0088
          START OF SEGMENT **********  106
                03421000 T  0089
                03422000 T  0089
                03423000 T  0089
                03424000 T  0089
                03425000 T  0089
                03426000 T  0089
```

```
 128,"DIV ",129,"DV2 ",130,"COM ",131,"LQV ",132,"SND ",133,"XCH ",              03427000 T  0089
 134,"CHS ",167,"RTS ",168,"CDC ",197,"FTC ",260,"LOD ",261,"DUP ",              03428000 T  0089
 278,"GBC ",280,"SSF ",294,"GFC ",322,"ZP1 ",384,"IDV ",453,"FTF ",              03429000 T  0089
 515,"MDS ",532,"ISD ",533,"LEQ ",534,"BBW ",548,"ISN ",549,"LSS ",              03430000 T  0089
 550,"BFW ",581,"EQL ",582,"SSP ",584,"ECM ",709,"CTC ",790,"GBW ",              03431000 T  0089
 806,"GFW ",896,"RDV ",965,"CTF ",                                               03432000 T  0089
        1023,1023,1023,1023,1023,1023,1023,1023,1023,1023,1023, 1023;            03433000 T  0089
                                                                    106 IS  132 LONG, NEXT SEG   97
FILL TIPE[*] WITH 10(-1),-19,-21,OCT300000000,-21,-2,-3,-4,                      03433100 T  0089
                                                                  START OF SEGMENT **********  107
                 8(OCT300000000),OCT100000000,-5,-13,-4,-6,-7,-19,-11,           03433110 T  0091
                 5(OCT100000000),-8,3(OCT300000000),-9,-10,-12,-13,-14,          03433120 T  0091
                 -15,-100,-16,8(OCT300000000),-17,-6,-18,-19,-20,-21 ;           03433130 T  0091
                                                                    107 IS   64 LONG, NEXT SEG   97
FILL PERIODWORD[*] WITH                                                          03434000 T  0091
   "FALSE ", "TRUE  ", "OR    ", "AND   ", "NOT   ",                             03435000 T  0091
                                                                  START OF SEGMENT **********  108
   "LT    ", "LE    ", "EQ    ", "GT    ", "GE    ", "NE    ";                   03436000 T  0092
                                                                    108 IS   11 LONG, NEXT SEG   97
ACCUM[0] ← EXACCUM[0] ← ";      ";                                               03437000 T  0092
INCLUDE := "NCLUDE" & "I"[6:42:6];                                               03437100 T  0095
INSERTDEPTH := -1;                                                               03437110 T  0096
FILL TEN[*] WITH   % POWERS OF TEN TO PRT 22                                     03438000 T  0097
    OCT1141000000000000,    OCT1131200000000000,    OCT1121440000000000,         03439000 T  0098
                                                                  START OF SEGMENT **********  109
    OCT1111750000000000,    OCT1102342000000000,    OCT1073032400000000,         03440000 T  0099
    OCT1063641100000000,    OCT1054611320000000,    OCT1045753604000000,         03441000 T  0099
    OCT1037346545000000,    OCT1011124027620000,    OCT0001351035564000,         03442000 T  0099
    OCT0011643245121000,    OCT0022214116345200,    OCT0032657142036440,         03443000 T  0099
    OCT0043432772446150,    OCT0054341571157602,    OCT0065432127413542,         03444000 T  0099
    OCT0076740555316473,    OCT0111053071060221,    OCT0121265707274265,         03445000 T  0099
    OCT0131543271153342,    OCT0142074147406233,    OCT0152513201307702,         03446000 T  0099
    OCT0163236041571663,    OCT0174105452130240,    OCT0205126764556310,         03447000 T  0099
    OCT0216354561711772,    OCT0231004771627437,    OCT0241206170175346,         03448000 T  0099
    OCT0251447626234640,    OCT0261761573704010,    OCT0272356132665012,         03449000 T  0099
    OCT0303051561442215,    OCT0313664115752660,    OCT0324641141345435,         03450000 T  0099
    OCT0336011371636744,    OCT0347413670206535,    OCT0361131664625026,         03451000 T  0099
    OCT0371360241772234,    OCT0401654312370703,    OCT0412227375067064,         03452000 T  0099
    OCT0422675274304701,    OCT0433454553366061,    OCT0444367706263475,         03453000 T  0099
    OCT0455465667740415,    OCT0467003245730520,    OCT0501060411731664,         03454000 T  0099
    OCT0511274514320241,    OCT0521553637404312,    OCT0532106607305374,         03455000 T  0099
    OCT0542530351166673,    OCT0553256443424452,    OCT0564132154331565,         03456000 T  0099
    OCT0575160607420123,    OCT0606414751324147,    OCT0621012014361120,         03457000 T  0099
    OCT0631214417455344,    OCT0641457523370635,    OCT0651773450267004,         03458000 T  0099
    OCT0662372362344605,    OCT0673071057035747,    OCT0703707272645341,         03459000 T  0099
    OCT0714671151416631,    OCT0726047403722377,    OCT0737461304707077,         03460000 T  0099
    OCT0751137556607071,    OCT0761367512350710,    OCT0771665435043072,         03461000 T  0099
    OCT0000000000000000,    OCT0000000000000000,    OCT0000000000000000,         03462000 T  0099
    OCT0000000000000000,    OCT0000000000000000,    OCT0000000000000000,         03463000 T  0099
    OCT0000000000000000,    OCT0000000000000000,    OCT0000000000000000,         03464000 T  0099
    OCT0000000000000000,    OCT0000000000000000,    OCT0000000000000000,         03465000 T  0099
    OCT0000000000000000,    OCT0000000000000000,    OCT0000000000000000,         03466000 T  0099
    OCT0000000000000000,    OCT0000000000000000,    OCT0004000000000000,         03467000 T  0099
    OCT0001000000000000,    OCT0001720000000000,    OCT0004304000000000,         03468000 T  0099
    OCT0007365000000000,    OCT0005262200000000,    OCT0004536640000000,         03469000 T  0099
    OCT0001666410000000,    OCT0000244112000000,    OCT0000315134400000,         03470000 T  0099
    OCT0000400363500000,    OCT0000450046042000,    OCT0006562057452400,         03471000 T  0099
```

```
    0CT0004316473365100,    0CT0005402212262320,    0CT0006702654737004,        03472000 T  0099
    0CT0004463430126605,    0CT0007600336154346,    0CT0001540425607437,        03473000 T  0099
    0CT0004070533151347,    0CT0005106662003641,    0CT0005033043640461,        03474000 T  0099
    0CT0002241654610575,    0CT0002712227752734,    0CT0001474675745524,        03475000 T  0099
    0CT0002014055337051,    0CT0004417070626663,    0CT0007522706774440,        03476000 T  0099
    0CT0003447470573550,    0CT0006361406732502,    0CT0005005571052122,        03477000 T  0099
    0CT0006207127264547,    0CT0001650755141700,    0CT0006223150372260,        03478000 T  0099
    0CT0007670002470733,    0CT0007646003207120,    0CT0005617404050743,        03479000 T  0099
    0CT0001163305063137,    0CT0007420166277771,    0CT0001732422375777,        03480000 T  0099
    0CT0002321127075377,    0CT0003005354714677,    0CT0005606650100057,        03481000 T  0099
    0CT0007150422120072,    0CT0003002526544103,    0CT0001603254275130,        03482000 T  0099
    0CT0004144127354356,    0CT0007175155247451,    0CT0007034410521363,        03483000 T  0099
    0CT0007664351264566,    0CT0003641443541723,    0CT0004611754472310;        03484000 T  0099
FILL INLINEINT[*] WITH % FILLS MUST BE IN ASCENDING ORDER FOR                   03484100 T  0099
                                                                  109 IS  138 LONG, NEXT SEG   97
                        % BINARY SEARCH IN FUNCTION AND DOITINLINE.             03484120 T  0100
                        % INLINEINT[1],[1:1] = 1 ONCE CODE FOR INTRINSIC        03484122 T  0100
                        %                        HAS BEEN EMITTED INLINE.       03484124 T  0100
                        % INLINEINT[1],[2:10]=INDEX INTO 2=ND WORD OF THE       03484126 T  0100
                        %                      CORR ENTRY IN INT.               03484128 T  0100
                        % INLINEINT[1],[12:36]=NAME OF INTRINSIC.               03484130 T  0100
%********FIRST FILL MUST BE NUMBER OF INTRINSICS ******************             03484140 T  0100
34,                                                                             03484160 T  0100
                                                                  START OF SEGMENT **********  110
"00ABS    ",                                                                    03484180 T  0101
"00AIMAG  ",                                                                    03484200 T  0101
"00AINT   ",                                                                    03484220 T  0101
"00AMAX0  ",                                                                    03484224 T  0101
"00AMAX1  ",                                                                    03484228 T  0101
"00AMIN0  ",                                                                    03484232 T  0101
"00AMIN1  ",                                                                    03484236 T  0101
"00AMOD   ",                                                                    03484240 T  0101
"00AND    ",                                                                    03484260 T  0101
"00CMPLX  ",                                                                    03484280 T  0101
"00COMPL  ",                                                                    03484300 T  0101
"00CONJG  ",                                                                    03484320 T  0101
"00DABS   ",                                                                    03484340 T  0101
"00DBLE   ",                                                                    03484360 T  0101
"00DIM    ",                                                                    03484380 T  0101
"00DSIGN  ",                                                                    03484400 T  0101
"00EQUIV  ",                                                                    03484420 T  0101
"00FLOAT  ",                                                                    03484440 T  0101
"00IABS   ",                                                                    03484460 T  0101
"00IDIM   ",                                                                    03484480 T  0101
"00IDINT  ",                                                                    03484500 T  0101
"00IFIX   ",                                                                    03484520 T  0101
"00INT    ",                                                                    03484540 T  0101
"00ISIGN  ",                                                                    03484560 T  0101
"00MAX0   ",                                                                    03484564 T  0101
"00MAX1   ",                                                                    03484568 T  0101
"00MIN0   ",                                                                    03484572 T  0101
"00MIN1   ",                                                                    03484576 T  0101
"00MOD    ",                                                                    03484580 T  0101
"000R     ",                                                                    03484600 T  0101
"00REAL   ",                                                                    03484620 T  0101
"00SIGN   ",                                                                    03484640 T  0101
"00SNGL   ",                                                                    03484660 T  0101
```

```
"OOTIME  ",                                                                    03484680 T  0101
 0 ;                                                                           03484990 T  0101
                                                                    110 IS   36 LONG, NEXT SEG   97
        FILL INT [*] WITH                                                      03485000 T  0101
COMMENT THESE NAMES (1-ST WORD OF EACH TWO-WORD ENTRY) MUST BE IN               03486000 T  0101
        ASCENDING ORDER FOR BINARY LOOKUPS.                                    03486010 T  0101
        THE SECOND WORD HAS THE FOLLOWING FORMAT:                              03486020 T  0101
        .[1:1] = 0 IF THE INTRINSIC DOES NOT HAVE A PERMANENT PRT              03486030 T  0101
                   LOCATION, OTHERWISE = 1.  MAY BE RESET BY                   03486040 T  0101
                   WRAPUP.  SEE .[18:6] BELOW.                                 03486050 T  0101
        .[2:1] = .INTSEEN = 1 IFF INTRINSICS FUNCTION HAS BEEN SEEN.           03486055 T  0101
        .[6:3] = .INTCLASS = CLASS OF THE INTRINSIC.                           03486060 T  0101
        .[9:3] = .INTPARMCLASS = CLASS OF PARAMETERS.                          03486070 T  0101
        .[12:6] = .INTINLINE = INDEX FOR DOITINLINE IF ≠0, OTHERWISE           03486080 T  0101
                               DO IT VIA INTRINSIC CALL.                       03486090 T  0101
        .[24:6] = .INTPRT = FIXED PRT LOCATION, SEE .[1:1] ABOVE.              03486100 T  0101
        .[30:6] = .INTPARMS = NUMBER OF PARAMETERS REQUIRED BY THE INT.        03486110 T  0101
        .[36:12] = .INTNUM = INTRINSICS NUMBER.                                03486120 T  0101
        THE FIELDS .[3:3] AND .[18:6] ARE SO FAR UNUSED.                       03486130 T  0101
;                                                                              03486140 T  0101
%                                                                              03486144 T  0101
%**************************************************************              03486145 T  0101
%*********  IF YOU ADD AN INTRINSIC, BE SURE TO CHANGE NUMINTM1  *******      03486146 T  0101
%*********  AT SEQUENCE NUMBER 00155211.....THANK YOU.           *******      03486147 T  0101
%**************************************************************              03486148 T  0101
%                                                                              03486149 T  0101
"ABS    ", OCT00330;0000010007,                                                03487000 T  0101
                                                          START OF SEGMENT **********  111
"AIMAG ", OCT00360;0000010074,                                                 03488000 T  0103
"AINT  ", OCT00330;0000010054,                                                 03489000 T  0103
"ALGAMA", OCT00330000000010127,                                                03490000 T  0103
"ALOG10", OCT00330000000010103,                                                03491000 T  0103
"ALOG  ", OCT20330000035010017,                                                03492000 T  0103
"AMAX0 ", OCT00312500000000031,                                                03493000 T  0103
"AMAX1 ", OCT00332500000000031,                                                03494000 T  0103
"AMIN0 ", OCT00312500000000032,                                                03495000 T  0103
"AMIN1 ", OCT00332500000000032,                                                03496000 T  0103
"AMOD  ", OCT00330400000020063,                                                03497000 T  0103
"AND   ", OCT00330500000020130,                                                03498000 T  0103
"ARCOS ", OCT00330000000010117,                                                03499000 T  0103
"ARSIN ", OCT20330000032010116,                                                03500000 T  0103
"ATAN2 ", OCT20330000044020114,                                                03501000 T  0103
"ATAN  ", OCT20330000037010016,                                                03501500 T  0103
"CABS  ", OCT20360000045010053,                                                03502000 T  0103
"CCOS  ", OCT00660000000010110,                                                03503000 T  0103
"CEXP  ", OCT00660000000010100,                                                03504000 T  0103
"CLOG  ", OCT00660000000010102,                                                03505000 T  0103
"CMPLX ", OCT00630600000020075,                                                03506000 T  0103
"COMPL ", OCT00330700000010132,                                                03507000 T  0103
"CONCAT", OCT00330000000050140,                                                03508000 T  0103
"CONJG ", OCT00661100000010076,                                                03509000 T  0103
"COSH  ", OCT00330000000010121,                                                03510000 T  0103
"COS   ", OCT00330000000010015,                                                03511000 T  0103
"COTAN ", OCT00330000000010112,                                                03512000 T  0103
"CSIN  ", OCT00660000000010106,                                                03513000 T  0103
"CSQRT ", OCT00660000000010124,                                                03514000 T  0103
"DABS  ", OCT00550100000010052,                                                03515000 T  0103
```

```
"DATAN2", OCT0055000000020115,                                  03516000 T  0103
"DATAN ", OCT2055000041010113,                                  03517000 T  0103
"DBLE  ", OCT0053120000010062,                                  03518000 T  0103
"DCOS  ", OCT0055000000010107,                                  03519000 T  0103
"DEXP  ", OCT2055000047010077,                                  03520000 T  0103
"DIM   ", OCT0033100000020072,                                  03521000 T  0103
"DLOG10", OCT0055000000010104,                                  03522000 T  0103
"DLOG  ", OCT2055000042010101,                                  03522500 T  0103
"DMAX1 ", OCT0055000000000066,                                  03523000 T  0103
"DMIN1 ", OCT0055000000000067,                                  03524000 T  0103
"DMOD  ", OCT2055000046020065,                                  03525000 T  0103
"DSIGN ", OCT0055130000020071,                                  03526000 T  0103
"DSIN  ", OCT2055000043010105,                                  03527000 T  0103
"DSQRT ", OCT2055000050010123,                                  03528000 T  0103
"EQUIV ", OCT0033140000020133,                                  03529000 T  0103
"ERF   ", OCT0033000000010125,                                  03530000 T  0103
"EXP   ", OCT2033000033010020,                                  03531000 T  0103
"FLOAT ", OCT0031150000010060,                                  03532000 T  0103
"GAMMA ", OCT2033000040010126,                                  03533000 T  0103
"IABS  ", OCT0011010000010007,                                  03534000 T  0103
"IDIM  ", OCT0011100000020072,                                  03535000 T  0103
"IDINT ", OCT0015240000010057,                                  03536000 T  0103
"IFIX  ", OCT0013030000010054,                                  03537000 T  0103
"INT   ", OCT0013030000010054,                                  03538000 T  0103
"ISIGN ", OCT0011160000020070,                                  03539000 T  0103
",ERR. ", OCT2000000030000134,                                  03540000 T  0103
",FBINB", OCT0000000000000160,                                  03540500 T  0103
",FINAM", OCT0000000000000154,                                  03541000 T  0103
",FONAM", OCT0000000000000155,                                  03542000 T  0103
",FREFR", OCT0000000000000146,                                  03543000 T  0103
",FREWR", OCT0000000000000153,                                  03544000 T  0103
",FTINT", OCT0000000000000050,                                  03545000 T  0103
",FTNIN", OCT0000000000000156,                                  03546000 T  0103
",FTNOU", OCT0000000000000157,                                  03547000 T  0103
",FTOUT", OCT0000000000000051,                                  03548000 T  0103
",LABEL", OCT0000000000000021,                                  03549000 T  0103
",MATH ", OCT0000000000000055,                                  03550000 T  0103
",MEMHR", OCT0000000000000164,                                  03550500 T  0103
",XTOI ", OCT0000000000000056,                                  03551000 T  0103
"MAXO  ", OCT0011250000000135,                                  03552000 T  0103
"MAX1  ", OCT0013250000000135,                                  03553000 T  0103
"MINO  ", OCT0011250000000136,                                  03554000 T  0103
"MIN1  ", OCT0013250000000136,                                  03555000 T  0103
"MOD   ", OCT0011170000020137,                                  03556000 T  0103
"OR    ", OCT0033200000020131,                                  03557000 T  0103
"REAL  ", OCT0036210000010073,                                  03558000 T  0103
"SIGN  ", OCT0033160000020070,                                  03559000 T  0103
"SINH  ", OCT0033000000010120,                                  03560000 T  0103
"SIN   ", OCT2033000034010014,                                  03561000 T  0103
"SNGL  ", OCT0035230000010061,                                  03562000 T  0103
"SQRT  ", OCT2033000031010013,                                  03563000 T  0103
"TANH  ", OCT0033000000010122,                                  03563010 T  0103
"TAN   ", OCT2033000036010111,                                  03563020 T  0103
"TIME  ", OCT0031220000010064,                                  03563030 T  0103
 0;                                                             03563900 T  0103
                                                        111 IS  169 LONG, NEXT SEG   97
BLANKS←INLINEINT[MAX←0] ;                                       03563910 T  0103
```

```
                FOR SCN←1 STEP 1 UNTIL BLANKS DO                                        03563920 T  0104
                    BEGIN                                                               03563930 T  0109
                    EQVID←INLINEINT[SCN]; WHILE INT[MAX]≠EQVID DO MAX←MAX+2 ;           03563940 T  0109
                    INLINEINT[SCN].INTX←MAX+1 ;                                         03563950 T  0113
                    END ;                                                               03563960 T  0116
                INTID.SUBCLASS ← INTYPE;                                                03564000 T  0118
                REALID.SUBCLASS ← REALTYPE;                                             03565000 T  0120
                FQVID ← ".EQ000";                                                       03566000 T  0122
                LISTID := ".LI000";                                                     03567100 T  0122
                BLANKS ← "      ";                                                      03568000 T  0123
                ENDSEGTOG ← TRUE;                                                       03569000 T  0124
                SCN ← 7;                                                                03570000 T  0126
                MAX ← REAL(NOT FALSE).[9:39];                                           03571000 T  0126
                SUPERMAXCOM←128×(MAXCOM+1) ;                                            03571100 T  0128
                SEGPTOG ← FALSE;   %INHIBIT PAGE SKIP AFTER SUBROUTINES      %501=      03571300 P  0130
                END   INITIALIZATION;                                                   03572000 T  0131
                                                                          97 IS  135 LONG, NEXT SEG    6
                ALPHA PROCEDURE NEED(T, C); VALUE T, C; ALPHA T, C;                     03573000 T  0506
                BEGIN INTEGER N;  REAL ELBAT;                                           03574000 T  0506
                                                                        START OF SEGMENT **********  112
STACK(F+3) = N
STACK(F+4) = ELBAT
                    REAL X;                                                             03574100 T  0000
STACK(F+5) = X
                    LABEL XIT, CHECK;                                                   03575000 T  0000
                    ALPHA INFA, INFB, INFC;                                             03576000 T  0000
STACK(F+6) = INFA
STACK(F+7) = INFB
STACK(F+10) = INFC
                COMMENT NEED RETURNS THE ELBAT WORD FOR THE IDENTIFIER T,               03577000 T  0000
IF THIS IS THE FIRST OCCURRENCE OF T THEN AN INFO WORD IS BUILT AND                     03578000 T  0000
GIVEN THEN CLASS C;                                                                     03579000 T  0000
   ELBAT.CLASS ← C;                                                                     03580000 T  0000
   XTA ← T;                                                                             03581000 T  0001
   IF C ≤ LABELID THEN                                                                  03582000 T  0002
      BEGIN                                                                             03583000 T  0003
         IF N ← SEARCH(T) = 0 THEN N ← ENTER(ELBAT, T) ELSE                             03584000 T  0003
         IF ELBAT ←  GET(N).CLASS = UNKNOWN                                             03585000 T  0007
           THEN PUT(N,GET(N)&C[TOCLASS])                                                03586000 T  0009
         ELSE IF ELBAT ≠ C THEN FLOG(21);                                               03587000 T  0013
         GO TO XIT;                                                                     03588000 T  0015
      END;                                                                              03589000 T  0016
   IF N ← SEARCH(T) = 0 THEN                                                            03590000 T  0016
      BEGIN                                                                             03591000 T  0018
         IF N ← GLOBALSEARCH(T) ≠ 0 THEN GO TO CHECK;                                   03592000 T  0018
         N ← GLOBALENTER(ELBAT, T);                                                     03593000 T  0020
         GO TO XIT;                                                                     03594000 T  0022
      END;                                                                              03595000 T  0022
   GETALL(N,INFA,INFB,INFC);                                                            03596000 T  0022
   IF INFA.CLASS = DUMMY THEN BEGIN N ← INFC.BASE; GO TO CHECK END;                     03596100 T  0024
   IF BOOLEAN(INFA   .FORMAL) THEN GO TO CHECK;                                         03597000 T  0027
   IF INFA.CLASS ≠ UNKNOWN THEN                                                         03598000 T  0029
   BEGIN                                                                                03599000 T  0030
      IF N ← GLOBALSEARCH(T) ≠ 0 THEN GO TO CHECK;                                      03600000 T  0031
      ELBAT.SUBCLASS ← INFA.SUBCLASS;                                                   03601000 T  0033
      N ← GLOBALENTER(ELBAT, T);                                                        03602000 T  0035
      GO TO XIT;                                                                        03603000 T  0037
```

```
                    END;
                    PUT(N, INFA & DUMMY[TOCLASS]);
                    ELBAT.SUBCLASS ← INFA   .SUBCLASS;
                    IF X ← GLOBALSEARCH(T) = 0 THEN X ← GLOBALENTER(ELBAT, T);
                    PUT(N+2, INFC & X[TOBASE]);  N ← X;
                    CHECK:
                    INFA ← GET(N);
                    IF ELBAT ← INFA   .CLASS = UNKNOWN THEN
                       BEGIN INFO[N,IR,N,IC].CLASS ← C; GO TO XIT END;
                    IF ELBAT ≠ C THEN
                       IF ELBAT = EXTID AND
                          (C = SUBRID OR C = FUNID) THEN
                          INFO[N,IR,N,IC].CLASS ← C
                       ELSE IF (ELBAT=SUBRID OR ELBAT= FUNID) AND C = EXTID THEN
                       ELSE FLOG(21);
                    XIT: NEED ← GETSPACE(N);
                    XTA ← NAME;     % RESTORE XTA FOR DIAGNOSTIC PURPOSES
                END NEED;

                                                                    112 IS   77 LONG, NEXT SEG    6
                INTEGER PROCEDURE EXPR(B); VALUE B; BOOLEAN B; FORWARD;
PRT(707) = EXPR
                PROCEDURE SPLIT(A); VALUE A; REAL A;
PRT(710) = SPLIT
                BEGIN
                      EMITPAIR(JUNK, ISN);
                      EMITD(40, DIA);
                      EMITD(18, ISO);
                      EMITDESCLIT(A);
                      EMITO(LOD);
                      EMITOPDCLIT(JUNK);
                      EMITPAIR(255,CHS) ;
                      EMITO(LND);
                END SPLIT;
                BOOLEAN PROCEDURE SUBSCRIPTS(LINK,FROM); VALUE LINK,FROM;
PRT(711) = SUBSCRIPTS
                INTEGER LINK, FROM;
                BEGIN INTEGER I, NSUBS, BDLINK;

                                                                  START OF SEGMENT **********  113
STACK(F+3) = I
STACK(F+4) = NSUBS
STACK(F+5) = BDLINK
                   LABEL CONSTRUCT, XIT;
                REAL SUM, PROD, BOUND;
STACK(F+6) = SUM
STACK(F+7) = PROD
STACK(F+10) = BOUND
                REAL INFA,INFB,INFC;
STACK(F+11) = INFA
STACK(F+12) = INFB
STACK(F+13) = INFC
                REAL SAVENSEG,SAVEADR ;
STACK(F+14) = SAVENSEG
STACK(F+15) = SAVEADR
                INTEGER  INDX;
STACK(F+16) = INDX
                REAL     INFD;
STACK(F+17) = INFD
```

```
03604000 T  0037
03605000 T  0037
03606000 T  0039
03607000 T  0041
03607100 T  0045
03608000 T  0048
03609000 T  0049
03610000 T  0050
03611000 T  0052
03612000 T  0057
03613000 T  0058
03614000 T  0059
03615000 T  0061
03616000 T  0064
03617000 T  0069
03618000 T  0071
03618100 T  0072
03619000 T  0073
03620000 T  0506
03621000 T  0506
03622000 T  0506
03623000 T  0506
03624000 T  0507
03625000 T  0508
03626000 T  0509
03627000 T  0509
03628000 T  0510
03629000 T  0511
03630000 T  0512
03631000 T  0513
03632000 T  0513
03633000 T  0513
03634000 T  0513
03635000 T  0000
03636000 T  0000
03637000 T  0000
03637100 T  0000
03637200 T  0000
03637300 T  0000
```

```
                  BOOLEAN TOG, VARF;                                                    03638000 T  0000
STACK(F+20) = TOG
STACK(F+21) = VARF
                  REAL SAVIT;                                                           03639000 T  0000
STACK(F+22) = SAVIT
                  DEFINE SS = LSTT#;                                                    03640000 T  0000
                                                                                        03641000 T  0000
                                                                                        03642000 T  0000
                  IF DEBUGTOG THEN FLAGROUTINE(" SUBSC","RIPTS ",TRUE ) ;               03643000 T  0000
                     SAVIT ← IT;                                                        03644000 T  0002
                     LINK ← GETSPACE(LINK);                                             03645000 T  0002
                  GETALL(LINK,INFA,INFB,INFC);                                          03646000 T  0004
                     IF INFA.CLASS ≠ ARRAYID THEN                                       03647000 T  0005
                        BEGIN XTA ← INFB; FLOG(35); GO TO XIT END;                      03648000 T  0006
                     NSUBS ← INFC.NEXTRA;                                               03649000 T  0012
                     IF FROM = 4 THEN                                                   03649100 T  0013
                       BEGIN IF NSUBS GTR SAVESUBS THEN SAVESUBS := NSUBS;              03649200 T  0014
                             IF NSUBS GTR NAMLIST[0] THEN NAMLIST[0] := NSUBS;          03649230 T  0016
                             NAMLIST[NAMEIND].[1:8] := NSUBS;                           03649250 T  0019
                             INFD := GET(NEED(".SUBAR",BLOCKID)),ADDR;                  03649300 T  0021
                       END;                                                             03649400 T  0024
                     BDLINK ← INFC.ADINFO-NSUBS+1;                                      03650000 T  0024
                     VARF ← INFC < 0;                                                   03651000 T  0026
                     FOR I ← 1 STEP 1 UNTIL NSUBS DO                                    03652000 T  0027
                     BEGIN                                                              03653000 T  0030
                        IT←IT+1;  SAVENSEG←NSEG;  SAVEADR←ADR ;                         03654000 T  0030
                        IF EXPR(TRUE) > REALTYPE THEN FLAG(98);                         03655000 T  0032
                        IF ADR=SAVEADR THEN FLAG(36) ;                                  03655500 T  0035
                        IF VARF THEN                                                    03656000 T  0037
                           IF EXPRESULT=NUMCLASS AND NSEG=SAVENSEG THEN                 03657000 T  0037
                        BEGIN                                                           03658000 T  0039
                           ADR←SAVEADR ;                                                03659000 T  0040
                           EMITNUM(EXPVALUE-1);                                         03660000 T  0041
                        END ELSE EMITPAIR(1, SUB)                                       03661000 T  0042
                     ELSE                                                               03662000 T  0043
                     IF EXPRESULT=NUMCLASS AND NSEG = SAVENSEG AND FROM NEQ 4 THEN      03663000 T  0043
                        BEGIN                                                           03664000 T  0047
                        ADR←SAVEADR;     IF SS[IT]←EXPVALUE≤0 THEN FLAG(154) ;          03664100 T  0047
                        END                                                             03664200 T  0051
                     ELSE SS[IT] ← @9;                                                  03665000 T  0051
                     IF FROM = 4 THEN                                                   03665010 T  0053
                        BEGIN IF VARF THEN BEGIN EMITO(DUP); EMITPAIR(1,ADD); END;      03665100 T  0053
                              EMITL(INDX); INDX := INDX+1;                              03665200 T  0056
                              EMITDESCLIT(INFD);                                        03665300 T  0058
                              EMITO(IF VARF THEN STD ELSE STN);                         03665400 T  0059
                        END;                                                            03665500 T  0061
                     IF I < NSUBS THEN                                                  03666000 T  0061
                     BEGIN                                                              03667000 T  0062
                        IF GLOBALNEXT ≠ COMMA THEN                                      03668000 T  0063
                           BEGIN XTA ← INFB; FLOG(23) END;                              03669000 T  0063
                        SCAN;                                                           03670000 T  0065
                     END;                                                               03671000 T  0066
                  END;                                                                  03672000 T  0066
                  IF GLOBALNEXT ≠ RPAREN THEN BEGIN XTA ← INFB; FLOG(24); END           03673000 T  0068
                     ELSE IF FROM < 2 THEN                                              03673100 T  0071
                        BEGIN SCAN; IF PREC > 0 THEN FROM ← 1; END;                     03673200 T  0073
                  SUM ← 0;                                                              03674000 T  0076
```

```
TOG ← VARF;                                                                    03675000 T  0077
IF VARF THEN                                                                   03676000 T  0078
FOR I ← NSUBS-1 STEP -1 UNTIL 1 DO                                             03677000 T  0078
BEGIN                                                                          03678000 T  0083
   IF BOUND ← EXTRAINFO[(BDLINK←BDLINK+1),IR,BDLINK,IC] < 0 THEN               03679000 T  0083
      EMITOPDCLIT(BOUND) ELSE EMITNUM(BOUND);                                  03680000 T  0087
   EMITO(MUL);                                                                 03681000 T  0090
   EMITO(ADD);                                                                 03682000 T  0090
END                                                                            03683000 T  0091
ELSE                                                                           03684000 T  0091
FOR I ← NSUBS STEP -1 UNTIL 1 DO                                               03685000 T  0092
BEGIN                                                                          03686000 T  0094
   IF I = 1 THEN BOUND ← 1 ELSE                                                03687000 T  0094
      BOUND ← EXTRAINFO[(BDLINK←BDLINK+1),IR,BDLINK,IC];                       03688000 T  0096
   IF T ← SS[SAVIT+I] < @9 THEN                                                03689000 T  0100
      BEGIN                                                                    03690000 T  0102
         SUM ← (SUM+T-1)×BOUND;                                                03691000 T  0102
         IF TOG THEN PROD ← PROD×BOUND;                                        03692000 T  0105
      END                                                                      03693000 T  0107
   ELSE                                                                        03694000 T  0107
   BEGIN                                                                       03695000 T  0107
      IF TOG THEN BEGIN EMITNUM(PROD); EMITO(MUL); EMITO(ADD) END              03696000 T  0109
      ELSE TOG ← TRUE;                                                         03697000 T  0112
      PROD ← BOUND;                                                            03698000 T  0113
      SUM ← (SUM-1)×BOUND;                                                     03699000 T  0114
   END;                                                                        03700000 T  0115
END;                                                                           03701000 T  0115
IF VARF THEN T ← @9;                                                           03702000 T  0118
IF INFA.SUBCLASS ≥ DOUBTYPE THEN                                               03703000 T  0119
BEGIN                                                                          03704000 T  0120
   IF TOG THEN                                                                 03705000 T  0121
   BEGIN                                                                       03706000 T  0121
      IF T < @9 THEN EMITNUM(2×PROD) ELSE EMITL(2);                            03707000 T  0122
      EMITO(MUL);                                                              03708000 T  0126
   END;                                                                        03709000 T  0127
   SUM ← SUM×2;                                                                03710000 T  0127
END ELSE                                                                       03711000 T  0128
IF T < @9 AND TOG THEN BEGIN EMITNUM(PROD); EMITO(MUL) END;                    03712000 T  0128
IF BOOLEAN(INFA.CE) THEN                                                       03713000 T  0132
   SUM ← SUM + INFC.BASE ELSE                                                  03714000 T  0133
   IF BOOLEAN(INFA.FORMAL) THEN                                                03715000 T  0135
      BEGIN EMITOPDCLIT(INFA.ADDR-1);                                          03716000 T  0137
         IF TOG THEN EMITO(ADD) ELSE TOG ← TRUE;                               03717000 T  0140
      END;                                                                     03718000 T  0142
IF BOOLEAN(INFA.TWOD) AND FROM >0 THEN                                         03719000 T  0142
BEGIN                                                                          03720000 T  0144
   IF SUM = 0 THEN                                                             03721000 T  0145
      IF TOG THEN ELSE                                                         03722000 T  0145
      BEGIN                                                                    03723000 T  0147
         EMITL(0);                                                             03724000 T  0147
         EMITDESCLIT(INFA.ADDR);                                               03725000 T  0148
         EMITO(LOD);                                                           03726000 T  0149
         EMITL(0);                                                             03727000 T  0150
         GO TO CONSTRUCT;                                                      03728000 T  0151
      END                                                                      03729000 T  0151
   ELSE                                                                        03730000 T  0151
      IF TOG THEN                                                              03731000 T  0151
```

```
            BEGIN                                                                    03732000 T  0152
               EMITNUM(ABS(SUM));                                                    03733000 T  0152
               IF SUM < 0 THEN EMITO(SUB) ELSE EMITO(ADD);                           03734000 T  0153
            END ELSE                                                                 03735000 T  0157
            BEGIN                                                                    03736000 T  0157
               EMITL(SUM,[33:7]);                                                    03737000 T  0157
               EMITDESCLIT(INFA,ADDR);                                               03738000 T  0158
               EMITO(LOD);                                                           03739000 T  0160
               EMITL(SUM,[40:8]);                                                    03740000 T  0160
               GO TO CONSTRUCT;                                                      03741000 T  0162
            END;                                                                     03742000 T  0162
         SPLIT(INFA,ADDR);                                                           03743000 T  0162
         CONSTRUCT;                                                                  03744000 T  0163
         IF BOOLEAN(FROM) THEN                                                       03745000 T  0164
         BEGIN                                                                       03746000 T  0164
            IF INFA.SUBCLASS ≥ DOUBTYPE THEN                                         03747000 T  0164
            BEGIN                                                                    03748000 T  0166
               EMITO(CDC);                                                           03749000 T  0166
               EMITO(DUP);                                                           03750000 T  0167
               EMITPAIR(1, XCH);                                                     03751000 T  0168
               EMITO(INX);                                                           03752000 T  0169
               EMITO(LOD);                                                           03753000 T  0169
               EMITO(XCH);                                                           03754000 T  0170
               EMITO(LOD);                                                           03755000 T  0171
            END ELSE EMITO(COC);                                                     03756000 T  0172
         END ELSE EMITO(CDC);                                                        03757000 T  0173
      END ELSE                                                                       03758000 T  0174
         BEGIN                                                                       03759000 T  0174
         IF SUM = 0 THEN IF NOT TOG THEN EMITL(0) ELSE                               03760000 T  0175
         ELSE                                                                        03761000 T  0178
         BEGIN                                                                       03762000 T  0178
            IF TOG THEN                                                              03763000 T  0179
               BEGIN                                                                 03764000 T  0179
                  EMITNUM(ABS(SUM));                                                 03765000 T  0179
                  IF SUM < 0 THEN EMITO(SUB) ELSE EMITO(ADD);                        03766000 T  0180
               END                                                                   03767000 T  0184
            ELSE EMITNUM(SUM);                                                       03768000 T  0184
         END;                                                                        03769000 T  0185
         IF FROM > 0 THEN                                                            03770000 T  0185
            IF BOOLEAN (FROM) THEN                                                   03771000 T  0186
               IF INFA.SUBCLASS ≥ DOUBTYPE THEN                                      03772000 T  0186
               BEGIN                                                                 03773000 T  0188
                  EMITDESCLIT(INFA,ADDR);                                            03774000 T  0189
                  EMITO(DUP);                                                        03775000 T  0190
                  EMITPAIR(1, XCH);                                                  03776000 T  0191
                  EMITO(INX);                                                        03777000 T  0192
                  EMITO(LOD);                                                        03778000 T  0192
                  EMITO(XCH);                                                        03779000 T  0193
                  EMITO(LOD);                                                        03780000 T  0194
               END ELSE EMITV(LINK) ELSE                                             03781000 T  0195
      BEGIN DESCREQ ← TRUE; EMITN(LINK); DESCREQ ← FALSE END;                        03782000 T  0196
      END;                                                                           03783000 T  0199
      XIT:                                                                           03784000 T  0199
      IT ← SAVIT;                                                                    03785000 T  0199
      SUBSCRIPTS ← BOOLEAN(FROM);                                                    03785100 T  0199
   IF DEBUGTOG THEN FLAGROUTINE(" SUBSC","RIPTS ",FALSE) ;                           03786000 T  0200
   END SUBSCRIPTS;                                                                   03787000 T  0202
```

```
                                                                                                                    113 IS  211 LONG, NEXT SEG    6
                BOOLEAN PROCEDURE BOUNDS(LINK); VALUE LINK; REAL LINK ;                                                  03788000 T  0513
PRT(712) = BOUNDS
                BEGIN                                                                                                    03789000 T  0513
                    COMMENT CALLED  TO PROCESS ARRAY BOUNDS;                                                             03790000 T  0513
                    BOOLEAN VARF, SINGLETOG;                                                                %109-        03791000 P  0513
                                                                                                          START OF SEGMENT **********  114
STACK(F+3) = VARF
STACK(F+4) = SINGLETOG
                   DEFINE FNEW = LINK#;                                                                                  03792000 T  0000
                    REAL T, NSUBS, INFA, INFB, INFC, FIRSTSS;                                                            03793000 T  0000
STACK(F+5) = T
STACK(F+6) = NSUBS
STACK(F+7) = INFA
STACK(F+10) = INFB
STACK(F+11) = INFC
STACK(F+12) = FIRSTSS
                    LABEL LOOP;                                                                                          03794000 T  0000
                                                                                                                         03795000 T  0000
                                                                                                                         03796000 T  0000
                IF DEBUGTOG THEN FLAGROUTINE("   BOU","NDS   ",TRUE ) ;                                              03797000 T  0000
                    GETALL(FNEW, INFA, INFB, INFC);                                                                      03798000 T  0002
                    FIRSTSS ← NEXTSS;                                                                                    03799000 T  0003
                   IF LINK < 0 THEN BEGIN SINGLETOG ← TRUE; LINK ← ABS(LINK) END;                           %109-        03799500 C  0004
                    LOOP:                                                                                                03800000 T  0007
                    IF NEXT = ID THEN                                                                                    03801000 T  0008
                    BEGIN                                                                                                03802000 T  0008
                         T ← GET(FNEXT ← GETSPACE(FNEXT));                                                               03802100 T  0009
                         IF T.CLASS ≠ VARID OR NOT BOOLEAN(T.FORMAL) THEN FLAG(92) ELSE                                  03803000 T  0011
                         IF T.SUBCLASS > REALTYPE THEN FLAG(93);                                                         03804000 T  0015
                         T ← -T.ADDR;                                                                                    03805000 T  0020
                         VARF ← TRUE;                                                                                    03806000 T  0022
                    END ELSE                                                                                             03807000 T  0022
                    IF NEXT = NUM THEN                                                                                   03808000 T  0022
                    BEGIN                                                                                                03809000 T  0024
                         IF NUMTYPE≠INTYPE THEN FLAG(113) ;                                                              03810000 T  0024
                         IF T←FNEXT=0 THEN FLAG(122) ;                                                                   03810010 T  0026
                         IF NOT VARF THEN IF NSUBS = 0 THEN LENGTH ← FNEXT ELSE                                          03812000 T  0029
                              LENGTH ← LENGTH×FNEXT;                                                                     03813000 T  0032
                    END ELSE FLOG(122);                                                                                  03814000 T  0033
                    EXTRAINFO[NEXTSS.IR,NEXTSS.IC] ← T;                                                                  03815000 T  0035
                    NEXTSS ← NEXTSS-1;                                                                                   03816000 T  0038
                    NSUBS ← NSUBS+1;                                                                                     03817000 T  0039
                    SCAN;                                                                                                03818000 T  0040
                    IF NEXT = COMMA THEN BEGIN SCAN; GO TO LOOP END;                                                     03819000 T  0041
                    IF NEXT ≠ RPAREN THEN FLOG(94);                                                                      03820000 T  0043
                    XTA ← INFB;                                                                                          03821000 T  0045
                    IF INFA.CLASS = ARRAYID THEN FLAG(95);                                                               03822000 T  0046
                    INFA.CLASS ← ARRAYID;                                                                                03823000 T  0048
                IF VARF THEN                                                                                             03827000 T  0050
                BEGIN                                                                                                    03828000 T  0050
                    IF NOT BOOLEAN(INFA.FORMAL) THEN FLAG(96);                                                           03829000 T  0051
                   IF NSUBS > 1 OR INFA      .SUBCLASS ≥ DOUBTYPE THEN                                                   03830000 T  0053
                      BEGIN BUMPLOCALS;LENGTH←LOCALS + 1536;BOUNDS←TRUE END ELSE                                         03831000 T  0055
                      LENGTH    ←-EXTRAINFO[FIRSTSS.IR,FIRSTSS.IC];                                                      03832000 T  0061
                END ELSE                                                                                                 03833000 T  0067
                 IF NOT SINGLETOG AND INFA.SUBCLASS > LOGTYPE THEN                                      %109-        03834000 P  0067
```

```
                        BEGIN LENGTH ← 2 × LENGTH; BOUNDS ← TRUE END;
                     IF LENGTH > 32767 THEN FLAG(99);
                     INFC ← LENGTH & NSUBS[TONEXTRA] & FIRSTSS[TOADINFO];
                    IF VARF THEN INFC ← -INFC;
                     PUT(FNEW, INFA); PUT(FNEW+2, INFC);
                     SCAN;
                IF DEBUGTOG THEN FLAGROUTINE("   BOU","NDS   ",FALSE) ;
                END BOUNDS;

                PROCEDURE PARAMETERS(LINK); VALUE LINK; REAL LINK;
PRT(713) = PARAMETERS
                BEGIN


                   LABEL LOOP;

                   REAL NPARMS, EX, INFC, PTYPE;
STACK(F+2) = NPARMS
STACK(F+3) = EX
STACK(F+4) = INFC
STACK(F+5) = PTYPE
                   ALPHA EXPNAME;
STACK(F+6) = EXPNAME
                   BOOLEAN CHECK, INTFID;
STACK(F+7) = CHECK
STACK(F+10) = INTFID
                   BOOLEAN NOTZEROP;
STACK(F+11) = NOTZEROP
                   REAL SAVIT;
STACK(F+12) = SAVIT
                   DEFINE PARMTYPE = LSTT#;
                   SAVIT ← IT ← IT+1;
                IF DEBUGTOG THEN FLAGROUTINE(" PARAM","ETERS ",TRUE ) ;
                   INFC ← GET(LINK+2);
                   IF CHECK ← BOOLEAN(INFC.[1:1]) THEN
                   BEGIN
                      EX ← INFC.ADINFO;
                      NOTZEROP ← INFC.NEXTRA ≠ 0;
                      INTFID ← INFC.[36:12] = 1;
                   END;
                   LOOP:
                   BEGIN SCAN;
                      EXPNAME ← NAME;
                      IF GLOBALNEXT = 0 AND NAME = "$      " THEN
                         BEGIN EXPRESULT ← LABELID; SCAN;
                            IF GLOBALNEXT ≠ NUM THEN FLAG(44);
                            EMITLABELDESC(NAME);
                            PTYPE ← 0;
                            SCAN;
                         END
                      ELSE PTYPE ← EXPR(CHECK AND EXTRAINFO[EX.IR,EX.IC],CLASS
                         = EXPCLASS AND INTFID);
                      IF EXPRESULT = NUMCLASS THEN
                         IF PTYPE = STRINGTYPE THEN
                         BEGIN
                            ADR ← ADR - 1;
                            PTYPE ← INTYPE;
```

```
%109-          03834500 C  0069
               03835000 T  0072
               03836000 T  0074
               03837000 T  0076
               03838000 T  0078
               03839000 T  0081
               03840000 T  0081
               03841000 T  0083
        114 IS   91 LONG, NEXT SEG    6
               03842000 T  0513
                     ↓
               03843000 T  0513
               03844000 T  0513
               03845000 T  0513
               03846000 T  0513
      START OF SEGMENT **********  115
               03847000 T  0000

               03848000 T  0000

               03849000 T  0000

               03850000 T  0000

               03851000 T  0000

               03852000 T  0000
               03853000 T  0000
               03854000 T  0001
               03855000 T  0003
               03856000 T  0005
               03857000 T  0006
               03858000 T  0007
               03859000 T  0008
               03859500 T  0010
               03860000 T  0012
               03861000 T  0012
               03862000 T  0012
               03863000 T  0012
               03864000 T  0013
               03866000 T  0015
               03867000 T  0016
               03868000 T  0018
               03869000 T  0019
               03870000 T  0020
               03871000 T  0020
               03872000 T  0020
               03873000 T  0027
               03874000 T  0030
               03875000 T  0031
               03876000 T  0032
               03876500 T  0032
               03877000 T  0034
```

```
                        EXPRESULT ← SUBSVAR;                                          03878000 T  0034
                        IF STRINGSIZE = 1 AND                                         03879000 T  0035
                           (T ← EXTRAINFO[EX,IR,EX,IC].CLASS = VARID OR               03880000 T  0036
                           T = EXPCLASS) THEN                                         03881000 T  0040
                        BEGIN                                                         03882000 T  0041
                           EXPRESULT ← EXPCLASS;                                      03883000 T  0041
                           EMITNUM(STRINGARRAY[0]);                                   03884000 T  0042
                        END ELSE                                                      03885000 T  0043
                        BEGIN                                                         03886000 T  0043
                        EXPRESULT←ARRAYID ;                                           03887000 T  0044
                        EMITPAIR(PRGDESCBLDR(1,0,0,NXAVIL←NXAVIL+1), LOD);            03888000 T  0044
                        EMITL(0);                                                     03889000 T  0048
                        WRITEDATA(STRINGSIZE, NXAVIL, STRINGARRAY);                   03890000 T  0048
                        END;                                                          03891000 T  0050
                     END ELSE EXPRESULT ← EXPCLASS;                                   03892000 T  0050
                  PARMTYPE[IT] ← 0 & EXPRESULT[TOCLASS] & PTYPE[TOSUBCL];             03893000 T  0051
                  XTA ← EXPNAME;                                                      03894000 T  0055
                  IF TSSEDITOG THEN IF (EXPRESULT=FUNID OR EXPRESULT=SUBRID OR        03894050 T  0055
                  EXPRESULT=EXTID) AND NOT DCINPUT THEN TSSED(XTA,2) ;                03894060 T  0058
                  IF DCINPUT THEN IF EXPRESULT=FUNID OR EXPRESULT=SUBRID              03894100 T  0062
                  OR EXPRESULT=EXTID THEN FLAG(151) ;                                 03894200 T  0064
                  IF CHECK THEN                                                       03895000 T  0067
                     BEGIN                                                            03896000 T  0068
                     IF T ← EXTRAINFO[EX,IR,EX,IC].CLASS ≠ EXPRESULT THEN             03897000 T  0068
                     CASE T OF                                                        03898000 T  0072
                     BEGIN                                                            03899000 T  0073
PRT(714) = *CASE STATEMENT DESCRIPTOR*                                                03900000 T  0073
                  EXTRAINFO[EX,IR,EX,IC] ← 0 & EXPRESULT[TOCLASS]                     03901000 T  0077
                     & PTYPE[TOSUBCL];                                                03902000 T  0079
                  IF EXPRESULT ≠ SUBSVAR THEN FLAG(66);                               03903000 T  0081
                  IF EXPRESULT = SUBSVAR THEN                                         03903100 T  0082
                     IF NOT INTFID THEN                                               03903150 T  0083
                     BEGIN  EMITO(CDC);                                               03903200 T  0084
                        IF PTYPE ≥ DOUBTYPE THEN EMITL(0);                            03903400 T  0086
                        END ELSE                                                      03903500 T  0086
                   ELSE                                                               03904000 T  0087
                     IF EXPRESULT = EXPCLASS THEN                                     03904100 T  0088
                        BEGIN IF PTYPE ≥ DOUBTYPE THEN EMITO(XCH);                    03904200 T  0090
                              EXTRAINFO[EX,IR,EX,IC].CLASS ← EXPCLASS                 03905000 T  0093
                        END ELSE FLAG(67);                                            03906000 T  0096
                  ; ; ;                                                               03907000 T  0096
                  FLAG(68);                                                           03908000 T  0098
                  IF EXPRESULT = EXTID THEN                                           03909000 T  0098
                     PUT(EXPLINK,GET(EXPLINK)&FUNID[TOCLASS]) ELSE                    03910000 T  0101
                     FLAG(69);                                                        03911000 T  0103
                  ;                                                                   03912000 T  0103
                  IF EXPRESULT = FUNID OR EXPRESULT = SUBRID THEN                     03913000 T  0105
                     EXTRAINFO[EX,IR,EX,IC] ← EXPRESULT ELSE FLAG(70);                03914000 T  0110
                  IF EXPRESULT = EXTID THEN                                           03915000 T  0111
                     PUT(EXPLINK,GET(EXPLINK)&SUBRID[TOCLASS]) ELSE                   03916000 T  0114
                     FLAG(71);                                                        03917000 T  0116
                  ; ;                                                                 03918000 T  0116
                  IF EXPRESULT = ARRAYID THEN EXTRAINFO[EX,IR,EX,IC].CLASS            03919000 T  0119
                     ← ARRAYID ELSE                                                   03920000 T  0121
                     IF EXPRESULT = VARID THEN                                        03921000 T  0122
                     BEGIN                                                            03922000 T  0123
                        EXTRAINFO[EX,IR,EX,IC].CLASS ← SBVEXP;
```

```
            EMITL(0)                                                                    03923000 T  0127
         END ELSE                                                                       03924000 T  0127
         IF EXPRESULT = EXPCLASS THEN                                                   03925000 T  0128
         BEGIN                                                                          03926000 T  0129
            EXTRAINFO[EX,IR,EX,IC].CLASS ← SBVEXP;                                      03927000 T  0130
            IF PTYPE ≥ DOUBTYPE THEN EMITO(XCH) ELSE EMITL(0);                          03928000 T  0134
         END ELSE FLAG(72);                                                             03929000 T  0137
      IF EXPRESULT = SUBSVAR THEN                                                       03930000 T  0139
         IF NOT INTFID THEN                                                             03930100 T  0140
            BEGIN EMITO(CDC);                                                           03930200 T  0141
                  IF PTYPE ≥ DOUBTYPE THEN EMITL(0)                                     03930300 T  0142
            END                                                                         03930400 T  0143
         ELSE                                                                           03930500 T  0144
        ELSE IF EXPRESULT = VARID THEN                                                  03930600 T  0144
             IF NOT INTFID THEN                                                         03930650 T  0146
            IF PTYPE ≥ DOUBTYPE THEN EMITL(0)  ELSE  ELSE                               03930700 T  0147
            ELSE FLAG(67);                                                              03930800 T  0150
      IF EXPRESULT = VARID THEN                                                         03931000 T  0152
         EMITL(0) ELSE                                                                  03932000 T  0153
      IF EXPRESULT = EXPCLASS THEN                                                      03933000 T  0154
         IF PTYPE ≥ DOUBTYPE THEN EMITO(XCH) ELSE EMITL(0)                              03934000 T  0155
      ELSE IF EXPRESULT ≠ SUBSVAR THEN FLAG(67);                                        03935000 T  0158
      END OF CASE STATEMENT                                                             03936000 T  0162
      ELSE IF PTYPE ≥ DOUBTYPE THEN                                                     03936100 T  0162
                                                                                  START OF SEGMENT **********  116
                                                                                   116 IS   17 LONG, NEXT SEG  115
           IF EXPRESULT = VARID THEN EMITL(0)                                           03936200 T  0164
           ELSE IF EXPRESULT = EXPCLASS AND NOT INTFID                                  03936300 T  0166
              THEN EMITO(XCH);                                                          03936400 T  0167
   IF T ← EXTRAINFO[EX,IR,EX,IC].SUBCLASS = 0 OR                                        03937000 T  0169
           (T = INTYPE AND PTYPE = REALTYPE AND                                         03938000 T  0173
           GET(LINK),SEGNO = 0) THEN                                                    03939000 T  0175
              EXTRAINFO[EX,IR,EX,IC].SUBCLASS ← PTYPE ELSE                              03940000 T  0177
           IF NOT(T = PTYPE OR T = REALTYPE AND PTYPE = INTYPE ) THEN                   03941000 T  0182
              FLAG(88);                                                                 03942000 T  0185
           END OF CHECK                                                                 03943000 T  0187
           ELSE IF PTYPE ≥ DOUBTYPE THEN                                                03943100 T  0187
                IF EXPRESULT = VARID THEN EMITL(0)                                      03943200 T  0188
                ELSE IF EXPRESULT = EXPCLASS THEN EMITO(XCH);                           03943300 T  0190
   IF NOTZEROP THEN EX ← EX+1;                                                          03944000 T  0193
      IT ← IT+1;                                                                        03945000 T  0195
   END;                                                                                 03946000 T  0196
   IF GLOBALNEXT = COMMA THEN GO TO  LOOP;                                              03947000 T  0196
   NPARMS ← IT - SAVIT;                                                                 03948000 T  0197
   IF GLOBALNEXT ≠ RPAREN THEN FLOG(108);                                               03949000 T  0199
   IF NOT CHECK THEN                                                                    03950000 T  0201
   BEGIN                                                                                03951000 T  0201
   INFC ← GET(LINK+2);                                                                  03952000 T  0202
   INFC ← -(INFC & NPARMS[TONFXTRA]                                                     03953000 T  0203
         & NEXTEXTRA[TOADINFO]);                                                        03954000 T  0205
   PUT(LINK+2,INFC);                                                                    03955000 T  0206
      FOR I ← SAVIT STEP 1 UNTIL IT-1 DO                                                03956000 T  0208
      BEGIN                                                                             03957000 T  0212
         EXTRAINFO[NEXTEXTRA,IR,NEXTEXTRA,IC] ← PARMTYPE[I];                            03958000 T  0212
         NEXTEXTRA ← NEXTEXTRA+1;                                                       03959000 T  0215
      END;                                                                              03960000 T  0217
   END                                                                                  03961000 T  0217
```

```
                    ELSE
                    IF T ← GET(LINK+2).NEXTRA > 0 AND T ≠ NPARMS OR                                  03962000 T  0217
                       T=0 AND INTFID AND NPARMS < 2 OR                                              03963000 T  0217
                       T = 0 AND  NOT INTFID THEN                                                    03964000 T  0221
                    BEGIN XTA ← GET(LINK+1); FLAG(28) END;                                           03964500 T  0224
                IF DEBUGTOG THEN FLAGROUTINE(" PARAM","ETERS ",FALSE) ;                              03965000 T  0226
                    IT ← SAVIT-1;                                                                    03966000 T  0229
                END PARAMETERS;                                                                      03967000 T  0231
                                                                                                     03968000 T  0232
                                                                                          115 IS  239 LONG, NEXT SEG    6
                PROCEDURE STMTFUNREF(LINK); VALUE LINK; REAL LINK;                                   03969000 T  0513
PRT(715) = STMTFUNREF
                BEGIN                                                                                03970000 T  0513
                        REAL I, PARMLINK, NPARMS, SEG;                                               03971000 T  0513
                                                                                                START OF SEGMENT **********  117
STACK(F+2) = I
STACK(F+3) = PARMLINK
STACK(F+4) = NPARMS
STACK(F+5) = SEG
                IF DEBUGTOG THEN FLAGROUTINE(" STMTF","UNREF ",TRUE ) ;                              03971010 T  0000
                        PARMLINK ← GET(LINK+2).[36:12];                                              03972000 T  0002
                        DO                                                                           03973000 T  0004
                     BEGIN                                                                           03974000 T  0005
                        SCAN;                                                                        03975000 T  0005
                        IF A←EXPR(TRUE) ≠ B←GET(PARMLINK).SUBCLASS THEN                              03976000 T  0005
                      IF A > REALTYPE OR B > REALTYPE THEN                              %108-        03977000 P  0008
                        BEGIN XTA ← NNEW; FLAG(88) END;                                              03978000 T  0011
                        PARMLINK ← PARMLINK-3;                                                       03979000 T  0013
                        NPARMS ← NPARMS+1;                                                           03980000 T  0014
                     END  UNTIL NEXT ≠ COMMA;                                                        03981000 T  0015
                        IF NEXT ≠ RPAREN THEN FLOG(108);                                             03982000 T  0016
                        SCAN;                                                                        03983000 T  0018
                        GETALL(LINK, INFA, XTA, INFC);                                               03984000 T  0019
                        IF NPARMS ≠ INFC.NEXTRA THEN FLAG(28);                                       03985000 T  0020
                        SEG ← INFA.SEGNO;                                                            03986000 T  0023
                        BRANCHLIT(INFC.BASE&SEG[TOSEGNO],FALSE);                                     03987000 T  0024
                        EMITB(INFA.ADDR & SEG[TOSEGNO], FALSE);                                      03988000 T  0027
                        ADJUST;                                                                      03989000 T  0029
                IF DEBUGTOG THEN FLAGROUTINE(" STMTF","UNREF ",FALSE) ;                              03989010 T  0030
                END STMTFUNREF;                                                                      03990000 T  0032
                                                                                          117 IS   37 LONG, NEXT SEG    6
                BOOLEAN PROCEDURE DOITINLINE(LNK);  VALUE LNK;  REAL LNK ;                           03990010 T  0513
PRT(716) = DOITINLINE
                    BEGIN                                                                            03990020 T  0513
                    REAL C,I,C1,C2,C3,C4,C5 ;                                                        03990030 T  0513
                                                                                                START OF SEGMENT **********  118
STACK(F+3) = C
STACK(F+4) = I
STACK(F+5) = C1
STACK(F+6) = C2
STACK(F+7) = C3
STACK(F+10) = C4
STACK(F+11) = C5
                    LABEL HUNT,FOUND,XIT,AIMAG,AINT,CMPLX,LOOP,DDT111,SNGL ;                         03990040 T  0000
                    DEFINE OPTYPE=LSTT#,  FO=FMITO#,  EP=EMITPAIR#,  EOL=EMITOPDCLIT# ;              03990045 T  0000
                    IF DEBUGTOG THEN FLAGROUTINE("DOITIN","LINE  ",TRUE) ;                           03990047 T  0000
                    C1←1;  C2←INLINEINT[0];  C3←GET(ABS(LNK)+1) ;                                    03990050 T  0002
                HUNT;                                                                                03990060 T  0005
```

```
        IF (C←INLINEINT[I←(C1+C2).[36:11]].INAM)<C3 THEN C1←I+1                    03990070 T  0006
        ELSE IF C>C3 THEN C2←I-1 ELSE GO FOUND ;                                  03990080 T  0010
        IF C1<C2 THEN GO HUNT ;                                                   03990082 T  0016
        IF INLINEINT[C1].INAM≠C3 THEN                                             03990084 T  0017
             BEGIN                                                                03990086 T  0019
             IF LNK<0 THEN DOITINLINE←BOOLEAN(LOOKFORINTRINSIC(-LNK)) ;            03990088 T  0019
             GO XIT ;                                                             03990090 T  0022
             END ;                                                                03990092 T  0023
        I←C1 ;                                                                    03990094 T  0023
    FOUND:                                                                        03990096 T  0023
        C1←(C←INT[C2←(C4←INLINEINT[I]).INTX]).INTPARMCLASS ;                       03990098 T  0024
        IF LNK<0 THEN                                                             03990100 T  0027
             BEGIN                                                                03990102 T  0028
             I←-LNK;  DOITINLINE←BOOLEAN(LNK←NEED(C3,FUNID)) ;                    03990104 T  0029
             IF (C2←GET(LNK+2))<0 THEN GO XIT ;                                   03990105 T  0032
             PUT(LNK+2,-(C2&(I←C.INTPARMS)[TONEXTRA]&NEXTEXTRA[TOADINFO])) ;       03990108 T  0034
             FOR I←I STEP -1 UNTIL 1 DO                                           03990112 T  0039
                  BEGIN                                                           03990114 T  0041
                  EXTRAINFO[NEXTEXTRA.IR.NEXTEXTRA.IC]←0&EXPCLASS[TOCLASS]        03990116 T  0041
                                                      &C1[TOSUBCL] ;              03990117 T  0044
                  NEXTEXTRA←NEXTEXTRA+1 ;                                         03990118 T  0046
                  END ;                                                           03990120 T  0047
             INFO[LNK.IR.LNK.IC].SUBCLASS←C.INTCLASS ;                            03990122 T  0049
             GO XIT ;                                                             03990124 T  0054
             END ;                                                                03990126 T  0054
        IF BOOLEAN(LNK.[2:1]) THEN                                                03990127 T  0054
        STACKHEAD[GET((NEXTINFO←NEXTINFO-3)+1) MOD SHX]←GET(NEXTINFO).LINK ;       03990128 T  0055
        LNK←C.INTINLINE;  INT[C2].INTSEEN←1;  DOITINLINE←TRUE ;                   03990130 T  0060
        IF C4>0 THEN INLINEINT[I]←-C4;  I←0 ;                                     03990132 T  0065
        IF XREF THEN ENTERX(C3,0&FUNID[TOCLASS]&C[21:6:3]);                       03990134 T  0068
             IF GLOBALNEXT≠LPAREN THEN BEGIN FLOG(106); GO XIT END ;              03990136 T  0072
        LOOP:  SCAN;  C5←XTA ;                                                    03990140 T  0075
             IF I=0 THEN                                                          03990145 T  0076
                  IF LNK=10 THEN EMITL(0) ELSE IF LNK=21 THEN EMITDESCLIT(2) ;    03990150 T  0077
             IF (C4←EXPR(TRUE))≠C1 AND (C1≠REALTYPE OR C4≠INTYPE) THEN            03990160 T  0082
                  BEGIN XTA←C5; FLAG(88); C2←-2 END ;                             03990165 T  0085
             I←I+1;  IF GLOBALNEXT=COMMA THEN GO LOOP ;                           03990170 T  0088
             IF GLOBALNEXT≠RPAREN THEN BEGIN FLOG(108); C2←-2 END;  SCAN ;        03990180 T  0091
             IF I≠C.INTPARMS THEN IF C.INTPARMS≠0 OR I<2 THEN                     03990190 T  0094
                  BEGIN XTA←C3;  FLAG(28);  C2←-2 END ;                           03990195 T  0098
             OPTYPE[IT]←C.INTCLASS;  IF C2<0 THEN GO XIT ;                        03990200 T  0101
             CASE (LNK-1) OF                                                      03990210 T  0104
                  BEGIN                                                           03990220 T  0105
PRT(717) = *CASE STATEMENT DESCRIPTOR*
                  EO(SSP) ; % @1: ABS, DABS, IABS,                                03990230 T  0106
             AIMAG:  EO(DEL) ; % @2: AIMAG,                                       03990240 T  0107
             AINT:   EP(1,IDV) ; % @3: AINT, IFIX, INT,                           03990250 T  0109
                  EO(RDV) ; % @4: AMOD,                                           03990260 T  0111
                  EO(LND) ; % @5: LOGICAL AND,                                    03990270 T  0112
             CMPLX:  EO(XCH) ; % @6: CMPLX,                                       03990280 T  0114
                  EO(LNG) ; % @7: LOGICAL COMPLIMENT (NEGATION),                  03990290 T  0115
                                                                                  03990291 T  0116
                  BEGIN % @10: DIM, IDIM,                                         03990300 T  0116
                  EO(SUB);  EO(DUP);  EP(0,LESS) ;                                03990310 T  0116
                  IF ADR>4082 THEN BEGIN ADR←ADR+1; SEGOVF END ;                  03990315 T  0119
                  EP(2,BFC);  EO(DEL);  EMITL(0) ;                                03990320 T  0122
                  END ;                                                           03990330 T  0124
```

```
                BEGIN EO(XCH); EO(CHS); GO CMPLX END ; % @11: CONJG,                                    03990331 T  0125
                ; % @12: DBLE (SOME CODE ALREADY EMITTED ABOVE),                                        03990340 T  0125
                                                                                                        03990350 T  0128
                                                                                                        03990351 T  0128
                BEGIN EO(XCH); EO(DEL) ; % @13: DSIGN,                                                  03990360 T  0128
DDT111:         EMITDDT(1,1,1) ;                                                                        03990370 T  0130
                END ;                                                                                   03990380 T  0131
                                                                                                        03990381 T  0131
                EO(LQV) ; % @14: LOGICAL EQUIVALENCE,                                                   03990390 T  0131
                ; % @15: FLOAT,                                                                        03990400 T  0133
                GO DDT111 ; % @16: ISIGN, SIGN,                                                         03990410 T  0133
                BEGIN EO(RDV); GO AINT END ; % @17: MOD,                                                03990420 T  0133
                EO(LOR) ; % @20: LOGICAL OR,                                                            03990430 T  0135
                BEGIN EO(XCH); GO AIMAG END ; % @21: REAL,                                              03990440 T  0136
                EP(1,KOM) ; % @22: TIMF,                                                                03990450 T  0138
                                                                                                        03990460 T  0139
                BEGIN % @23: SNGL,                                                                      03990470 T  0139
SNGL:           EP(9,SND); EO(XCH); EMITDDT(47,9,1); EMITL(0) ;                                         03990480 T  0139
                EMITDDT(9,9,38); EOL(9); EMITO(ADD); IF LNK=20 THEN GO AINT ;                           03990490 T  0143
                END ;                                                                                   03990500 T  0147
                                                                                                        03990510 T  0148
                GO SNGL ; % @24: IDINT,                                                                 03990520 T  0148
                                                                                                        03990530 T  0148
                BEGIN % @25: AMAX0,AMAX1,AMIN0,AMIN1,MAX0,MAX1,MIN0,MIN1,                               03990535 T  0148
                      %      SOME CODE ALREADY EMITTED ABOVE,                                           03990540 T  0148
                IF ADR>4068 THEN BEGIN ADR←ADR+1; SEGOVF END ;                                          03990542 T  0148
                EP(9,STD); EO(DUP); EOL(9) ;                                                            03990545 T  0151
                EO(IF C3,[24:6]="A" OR C3,[24:6]="X" THEN LESS ELSE GRTR) ;                             03990550 T  0154
                EP(2,BFC); EO(DEL); EOL(9); EO(XCH); EO(TOP); EO(LNG) ;                                 03990555 T  0159
                EP(14,BBC); EO(DEL); IF C3="MIN1  " OR C3="MAX1  " THEN GO AINT                         03990560 T  0163
                END ;                                                                                   03990565 T  0167
                                                                                                        03990566 T  0168
                END OF CASE STATEMENT ;                                                                 03990800 T  0168
                                                                                                      START OF SEGMENT **********  119
                                                                                                       119 IS   22 LONG, NEXT SEG  118
            XIT:                                                                                        03990810 T  0169
                IF DEBUGTOG THEN FLAGROUTINE("DOITIN","LINE  ",FALSE) ;                                 03990815 T  0169
                END OF DOITINLINE ;                                                                     03990820 T  0171
                                                                                                       118 IS  180 LONG, NEXT SEG    6
            REAL PROCEDURE LOOKFORINTRINSIC(L); VALUE L; REAL L;                                        03991000 T  0513
            BEGIN                                                                                       03992000 T  0513
               ALPHA ID, I, X, NPARMS;                                                                  03993000 T  0513
                                                                                                      START OF SEGMENT **********  120
STACK(F+3) = ID
STACK(F+4) = I
STACK(F+5) = X
STACK(F+6) = NPARMS
                REAL T;                                                                                 03994000 T  0000
STACK(F+7) = T
               LABEL FOUND, XIT;                                                                        03995000 T  0000
            IF DEBUGTOG THEN FLAGROUTINE("LOOKFO","RINTRN",TRUE ) ;                                     03995010 T  0000
               LOOKFORINTRINSIC ← L ← NEED(ID ← GET(L+1),FUNID);                                        03996000 T  0002
               IF GET(L+2) < 0 THEN GO TO XIT; % PARAMETER INFO KNOWN                                   03996050 T  0005
               COMMENT B MUST BE SET TO K/2, WHERE K IS THE INDEX OF THE LAST                           03996100 T  0007
               INTRINSIC NAME IN THE ARRAY INT;                                                         03996200 T  0007
                     A←0;  B←NUMINTM1 ;                                                                 03997000 T  0007
                     WHILE A+1 < B DO                                                                   03998000 T  0009
```

```
                    BEGIN
                       I ← REAL(BOOLEAN(A+B) AND BOOLEAN(1022));
                       IF Z ← INT[I] = ID THEN GO TO FOUND;
                       IF ID < Z THEN B ← I,[36:11] ELSE A ← I,[36:11];
                    END;
                    IF ID = INT[I←(A+B)×2-1] THEN GO TO FOUND;
                 GO TO XIT;
                 FOUND:
                  NPARMS←(X←INT[I+1]),INTPARMS;  INT[I+1],INTSEEN←1 ;
                  INFO[L,IR←L,IC],SUBCLASS←X,INTCLASS ;
                 PUT(L+2,-(1&NEXTEXTRA[TOADINFO]&NPARMS[TONEXTRA]));
                 IF NPARMS = 0 THEN NPARMS ← 1;
                  T←X,INTPARMCLASS ;
                 FOR I ← 1 STEP 1 UNTIL NPARMS DO
                 BEGIN
                    EXTRAINFO[NEXTEXTRA,IR←NEXTEXTRA,IC] ←
                       0 & EXPCLASS[TOCLASS] & T[TOSUBCL];
                    NEXTEXTRA ← NEXTEXTRA + 1;
                 END;
                 XIT:
              IF DEBUGTOG THEN FLAGROUTINE("LOOKFO","RINTRN",FALSE) ;
              END LOOKFORINTRINSIC;

              INTEGER PROCEDURE EXPR(VALREQ); VALUE VALREQ; BOOLEAN VALREQ;
              BEGIN LABEL LOOP, STACK, XIT, NOSCAN; REAL T;

STACK(F+3) = T
                 LABEL ARRY;
              LABEL HERE ;

                    REAL SAVIT, SAVIP;
STACK(F+4) = SAVIT
STACK(F+5) = SAVIP
                    BOOLEAN CNSTSEENLAST;       %FOR HANDLING CONSTANT          %113-
STACK(F+6) = CNSTSEENLAST
                    REAL SAVEADR;               %EXPONENTS                      %113-
STACK(F+7) = SAVEADR
                    DEFINE OPTYPE = LSTT#;
                 REAL EXPRESLT,EXPLNK;
STACK(F+10) = EXPRESLT
STACK(F+11) = EXPLNK
                    REAL EXPV;
STACK(F+12) = EXPV
                     REAL TM ;
STACK(F+13) = TM
                     DEFINE EO=EMITOP#, EP=EMITPAIR#, EOL=EMITOPNOLIT#,
                            ES1(OP)=BEGIN EO(XCH); EP(9,STD); EO(OP) END #,
                            ES2=BEGIN  EP(9,STD); EO(XCH);  EP(17,SND); EO(MUL) END # ;
                     LABEL CTYP, DTYP, RLESSC, DLESSC, CLESSD, CLESSC, RPLUSD, DTIMESC,
                           CTIMESR, CDIVBYD, CTIMESR1, CTIMESR2, DLESSC1 ;
                     LABEL SPECCHAR, RELATION;
                    REAL LINK;
STACK(F+14) = LINK
                    DEFINE T1 = EXPT1#, T2 = EXPT2#, CODE = EXPT3#;
                 COMMENT THE FOLLOWING TABLE GIVES THE PRECEDENCE (PREC) AND
                 OPERATOR NUMBER (OP) OF THE ARITHMETIC AND LOGICAL OPERATORS,
```

```
03999000 T  0011
04000000 T  0011
04001000 T  0013
04002000 T  0015
04003000 T  0022
04004000 T  0022
04005000 T  0026
04006000 T  0026
04007000 T  0027
04008000 T  0032
04009000 T  0037
04010000 T  0041
04011000 T  0043
04012000 T  0044
04013000 T  0045
04014000 T  0045
04015000 T  0047
04016000 T  0050
04017000 T  0051
04018000 T  0053
04018010 T  0054
04019000 T  0056
 120 IS   62 LONG, NEXT SEG    6
04020000 T  0513
04021000 T  0513
START OF SEGMENT **********  121
04022000 T  0000
04022010 T  0000
04023000 T  0000
04024000 T  0000
04025000 T  0000
04025500 C  0000
04025600 C  0000
04026000 T  0000
04027000 T  0000
04028000 T  0000
04028010 T  0000
04028020 T  0000
04028030 T  0000
04028040 T  0000
04028050 T  0000
04028060 T  0000
04029000 T  0000
04030000 T  0000
04031000 T  0000
04032000 T  0000
04033000 T  0000
```

```
                        OPERATOR            PREC      OP                                  04034000 T  0000
                        **                  9         15                                  04035000 T  0000
                        UNARY -             8         12                                  04036000 T  0000
                        /                   7         14                                  04037000 T  0000
                        *                   7         13                                  04038000 T  0000
                        -                   5         11                                  04039000 T  0000
                        +                   5         10                                  04040000 T  0000
                        .NE.                4         9                                   04041000 T  0000
                        .GE.                4         8                                   04042000 T  0000
                        .GT.                4         7                                   04043000 T  0000
                        .EQ.                4         6                                   04044000 T  0000
                        .LE.                4         5                                   04045000 T  0000
                        .LT.                4         4                                   04046000 T  0000
                        .NOT.               3         3                                   04047000 T  0000
                        .AND.               2         2                                   04048000 T  0000
                        .OR.                1         1                                   04049000 T  0000
                        THE UNARY PLUS IS IGNORED;                                        04050000 T  0000
                    PROCEDURE MATH(D, C, T); VALUE D, C, T; REAL D, C, T;                 04051000 T  0000
PRT(720) = MATH
                    BEGIN                                                                 04052000 T  0000
                       EMITO(MKS);                                                        04053000 T  0000
                       EMITL(C);                                                          04054000 T  0000
                       EMITV(NEED(",MATH ", INTRFUNID));                                  04055000 T  0001
                       EMITO(DEL);                                                        04056000 T  0003
                       IF D = 2 THEN EMITO(DEL);                                          04057000 T  0003
                       OPTYPE[IT←IT-1] ← T;                                               04058000 T  0005
                    END MATH;                                                             04059000 T  0008
                       NNEW ← NAME;                                                       04060000 T  0010
                    IF DEBUGTOG THEN FLAGROUTINE(" EXPRE","SSION ",TRUE ) ;               04061000 T  0010
                       OPTYPE[SAVIT←IT ← IT+1] ←                                          04062000 T  0012
                    PR[SAVIP←IP←IP+1] ← OPST[IP] ← 0;                                     04063000 T  0014
                    IF GLOBALNEXT = PLUS THEN GO TO LOOP;                                 04064000 T  0019
                    IF GLOBALNEXT = MINUS THEN                                            04065000 T  0020
                    BEGIN PREC ← 8; OP ← 12; GO TO STACK END;                             04066000 T  0021
                    IF PREC > 0 THEN GO TO STACK;                                         04067000 T  0026
                    LINK←(EXPLNK←FNEXT)&REAL(SCANENTER)[2:47:1] ;                         04068000 T  0027
                    GO TO NOSCAN;                                                         04069000 T  0029
                    LOOP: SCAN;                                                           04070000 T  0030
                    LINK ← FNEXT;                                                         04071000 T  0030
                    NOSCAN:                                                               04072000 T  0031
                    CNSTSEENLAST:=FALSE;                                    %113=         04072500 C  0032
                    IF GLOBALNEXT = ID THEN                                               04073000 T  0032
                    BEGIN                                                                 04074000 T  0033
                       IF IP ≠ SAVIP THEN EXPRESLT ← EXPCLASS;                            04074100 T  0034
                       OPTYPE[IT←IT+1] ← (A←GET(LINK)).SUBCLASS;                          04075000 T  0036
                       SCAN;                                                              04076000 T  0039
                       IF NOT RANDOMTOG THEN                                              04076050 T  0040
                       IF NEXT=EQUAL THEN BEGIN NEXT←0; OP←6; PREC←4 END ;                04076100 T  0041
                       IF GLOBALNEXT = ID OR GLOBALNEXT = NUM THEN                        04077000 T  0045
                          BEGIN FLOG(1); GO TO XIT END;                                   04078000 T  0047
                       IF NOT VALREQ AND PREC > 0 THEN VALREQ ← TRUE;                     04078100 T  0048
                       IF GLOBALNEXT ≠ LPAREN THEN                                        04079000 T  0051
                       BEGIN                                                              04080000 T  0052
                          LINK ← GETSPACE(LINK);                                          04081000 T  0052
                          T ← (A←GET(LINK)).CLASS;                                        04082000 T  0054
                          IF XREF THEN ENTERX(GET(LINK+1),0&A[15:15:9]);                  04082100 T  0056
                          IF EXPRESLT = 0 THEN EXPRESLT ← T;                              04083000 T  0060
```

```
      IF VALREQ THEN                                                                  04084000 T  0062
         IF T = VARID THEN EMITV(LINK) ELSE                                           04085000 T  0062
            BEGIN XTA ← GET(LINK+1); FLAG(50) END                                     04086000 T  0064
      ELSE                                                                            04087000 T  0067
      BEGIN                                                                           04088000 T  0067
         IF T = VARID THEN                                                            04089000 T  0068
            IF GLOBALNEXT > SLASH AND EXPRESLT = VARID THEN                           04090000 T  0069
            BEGIN                                                                     04091000 T  0071
               DESCREQ ← TRUE; EMITN(LINK); DESCREQ ← FALSE;                          04092000 T  0071
               GO TO XIT;                                                             04093000 T  0074
            END ELSE EMITV(LINK)                                                      04094000 T  0074
         ELSE                                                                         04095000 T  0075
         BEGIN                                                                        04096000 T  0075
            IF T = ARRAYID THEN                                                       04097000 T  0076
            BEGIN                                                                     04098000 T  0077
               IF BOOLEAN(A.CE) THEN                                                  04099000 T  0077
                  EMITNUM(GET(LINK+2),BASE) ELSE                                      04100000 T  0078
                  IF BOOLEAN(A,FORMAL) THEN                                           04101000 T  0081
                    EMITOPDCLIT(A.ADDR-1) ELSE                                        04102000 T  0082
                   EMITL(0);                                                          04103000 T  0084
               GO TO ARRY;                                                            04104000 T  0086
            END ELSE EMITPAIR(A.ADDR,LOD);                                            04105000 T  0086
            GO TO XIT;                                                                04106000 T  0088
         END;                                                                         04107000 T  0089
      END;                                                                            04108000 T  0089
      GO TO SPECCHAR;                                                                 04109000 T  0089
   END;                                                                               04110000 T  0089
   IF A.CLASS ≠ ARRAYID THEN                                                          04111000 T  0089
   BEGIN COMMENT FUNCTION REFERENCE;                                                  04112000 T  0090
      EXPRESLT ← EXPCLASS;                                                            04112100 T  0091
      IF A.CLASS = STMTFUNID THEN                                                     04113000 T  0092
        BEGIN                                                                         04114000 T  0093
        IF XREF THEN ENTERX(GET(LINK+1),0&A[15:15:9]);                                04114050 T  0093
        STMTFUNREF(LINK) ;                                                            04114100 T  0097
        IF NEXT=EQUAL THEN IF NOT RANDOMTOG THEN                                      04114200 T  0098
             BEGIN NEXT←0; OP←6; PREC←4 END ;                                         04114300 T  0100
        GO TO SPECCHAR ;                                                              04114400 T  0103
        END-;                                                                         04114500 T  0103
      IF A.CLASS=EXTID OR GET(TM←GLOBALSEARCH(GET(LINK+1))).CLASS=                    04114520 T  0103
        EXTID THEN LINK←REAL(DOITINLINE(-LINK)) ELSE                                  04114530 T  0108
      IF A.CLASS<FUNID AND TM=0 THEN                                                  04115000 T  0111
        IF DOITINLINE(LINK) THEN GO HERE ELSE                                         04115010 T  0113
         LINK ← LOOKFORINTRINSIC(LINK) ELSE                                           04116000 T  0115
         LINK ← NEED(GET(LINK+1),FUNID);                                              04117000 T  0117
       IF XREF THEN ENTERX(GET(LINK+1),0&GET(LINK)[15:15:9]);                         04117100 T  0120
      EMITO(MKS);                                                                     04118000 T  0124
                                                                                      04119000 T  0125
      PARAMETERS(LINK);                                                               04120000 T  0125
      IF ERRORTOG THEN GO TO XIT ELSE SCAN;                                           04121000 T  0125
      EMITV(LINK );                                                                   04122000 T  0127
      IF OPTYPE[IT] ← GET(LINK).SUBCLASS ≥ DOUBTYPE THEN                              04123000 T  0128
         EMITOPDCLIT(JUNK);                                                           04124000 T  0130
HERE; IF NEXT=EQUAL THEN                                                              04124100 T  0132
        IF NOT RANDOMTOG THEN BEGIN NEXT←0; PREC←4; OP←6 END ;                        04124200 T  0132
   END ELSE                                                                           04125000 T  0137
   BEGIN COMMENT ARRAY REFERENCE;                                                     04126000 T  0137
      IF XREF THEN ENTERX(GET(LINK+1),0&A[15:15:9]);                                  04126100 T  0137
```

```
SCAN;                                                                  04127000 T  0141
VALREQ ← SUBSCRIPTS(LINK,REAL(VALREQ));                                04128000 T  0141
IF ERRORTOG THEN GO TO XIT;                                            04129000 T  0143
IF NEXT=EQUAL THEN IF NOT RANDOMTOG THEN                               04129100 T  0144
     BEGIN NEXT←0; OP←6; PREC←4 END ;                                  04129200 T  0146
IF EXPRESLT  = 0 THEN EXPRESLT  ← SUBSVAR;                             04130000 T  0149
IF NOT VALREQ THEN                                                     04131000 T  0151
BEGIN                                                                  04132000 T  0151
   IF GLOBALNEXT > SLASH AND EXPRESLT = SUBSVAR THEN                   04133000 T  0152
   BEGIN                                                               04134000 T  0154
      ARRY;                                                            04135000 T  0154
      IF BOOLEAN((A←GET(LINK)),TWOD) THEN                              04136000 T  0155
      BEGIN                                                            04137000 T  0156
         EMITPAIR(TWODPRT, LOD);                                       04138000 T  0157
         T ← A.ADDR;                                                   04139000 T  0161
         IF T ≤ 1023 THEN                                              04140000 T  0163
         BEGIN                                                         04141000 T  0163
            EMITL(T,[38:10]);                                          04142000 T  0164
            EMITDESCLIT(10);                                           04143000 T  0165
         END ELSE                                                      04144000 T  0166
         BEGIN                                                         04145000 T  0166
            EMITL(T,[40:8]);                                           04146000 T  0166
            EMITDESCLIT(1536);                                         04147000 T  0168
            EMITO(INX);                                                04148000 T  0168
         END;                                                          04149000 T  0169
         EMITO(CTF);                                                   04150000 T  0169
      END ELSE EMITPAIR(A.ADDR,LOD);                                   04151000 T  0170
      EMITO(XCH);                                                      04152000 T  0173
      GO TO XIT;                                                       04153000 T  0174
      END;                                                             04154000 T  0174
      IF BOOLEAN((A←GET(LINK)),TWOD) THEN                              04155000 T  0174
      BEGIN                                                            04156000 T  0176
         SPLIT(A.ADDR);                                                04157000 T  0177
         IF A.SUBCLASS ≥ DOUBTYPE THEN                                 04158000 T  0178
         BEGIN                                                         04159000 T  0179
            EMITO(CDC);                                                04160000 T  0180
            EMITO(DUP);                                                04161000 T  0180
            EMITPAIR(1, XCH);                                          04162000 T  0181
            EMITO(INX);                                                04163000 T  0182
            EMITO(LOD);                                                04164000 T  0183
            EMITO(XCH);                                                04165000 T  0184
            EMITO(LOD);                                                04166000 T  0184
         END ELSE EMITO(COC);                                          04167000 T  0185
      END ELSE                                                         04168000 T  0186
      EMITV(LINK);                                                     04169000 T  0186
   END;                                                                04170000 T  0188
   END ARRAY REFERENCE;                                                04171000 T  0188
   GO TO SPECCHAR;                                                     04172000 T  0188
END;                                                                   04173000 T  0188
IF GLOBALNEXT = NUM THEN                                               04174000 T  0188
BEGIN                                                                  04175000 T  0189
   IF NUMTYPE = STRINGTYPE THEN                                        04176000 T  0189
      IF VALREQ THEN                                                   04177000 T  0190
      BEGIN                                                            04177200 T  0191
      NUMTYPE←INTYPE ;                                                 04177400 T  0191
      IF STRINGSIZE=1 THEN FNEXT←STRINGARRAY[0]                        04177500 T  0192
      ELSE BEGIN                                                       04177550 T  0194
```

```
            IF STRINGSIZE>2 OR STRINGARRAY[1].[18:30]≠"      " THEN
               FLAG(162) ;
            IF (FNEXT←STRINGARRAY[1].[12:6]&STRINGARRAY[0][6:12:36])
              .[6:6]>7 THEN NUMTYPE←REALTYPE ;
            END ;
         END;
SAVEADR:=ADR; CNSTSEENLAST:=TRUE;                                        %113-
   IF NUMTYPE = DOUBTYPE THEN
      EMITNUM2(FNEXT.DBLOW) ELSE EMITNUM (FNEXT);
   OPTYPE[IT←IT+1] ← NUMTYPE;
   IF EXPRESLT  = 0 THEN
   BEGIN EXPRESLT  ← NUMCLASS; EXPV      ← FNEXT END;
   SCAN;
   IF NOT RANDOMTOG THEN
   IF NEXT=EQUAL THEN BEGIN NEXT←0; OP←6; PREC←4 END ;
   IF NOT VALREQ AND PREC > 0 THEN VALREQ ← TRUE;
   IF GLOBALNEXT = ID OR GLOBALNEXT = NUM THEN
      BEGIN FLOG(1); GO TO XIT END;
END;
SPECCHAR:
IF GLOBALNEXT = LPAREN THEN
BEGIN
   SCAN;
   OPTYPE[IT←IT+1] ← EXPR(TRUE);

   IF GLOBALNEXT = COMMA AND EXPRESULT = NUMCLASS THEN
      BEGIN
         IF OPTYPE[IT] > REALTYPE THEN FLAG(85);
         SCAN;
         IF EXPR(TRUE) > REALTYPE
            OR EXPRESULT ≠ NUMCLASS THEN FLAG(85);
         EMITO(XCH);
         OPTYPE[IT] ← COMPTYPE;
         IF EXPRESLT = 0 THEN EXPRESLT ← NUMCLASS;
      END ELSE EXPRESLT ← EXPCLASS;
   IF GLOBALNEXT ≠ RPAREN THEN
   BEGIN FLOG(108); GO TO XIT END;
   GO TO LOOP;
END;
   WHILE PR[IP] ≥ PREC DO
BEGIN
   IF IT ≤ SAVIT THEN GO TO XIT;
   CODE ← MAP[T1←OPTYPE[IT-1]]×3 + MAP[T2←OPTYPE[IT]];
   CASE OPST[IP] OF
BEGIN
PRT(721) = *CASE STATEMENT DESCRIPTOR*
   GO TO XIT;
   BEGIN
      IF T1 = LOGTYPE AND T2 = LOGTYPE THEN EMITO(LOR)
         ELSE FLAG(51);
      IT ← IT-1;
   END;
   BEGIN
      IF T1 = LOGTYPE AND T2 = LOGTYPE THEN EMITO(LND)
         ELSE FLAG(52);
      IT ← IT-1;
   END;
```

```
04177575 T  0195
04177600 T  0197
04177700 T  0199
04177800 T  0201
04177900 T  0204
04178000 T  0204
04178500 C  0204
04179000 T  0205
04180000 T  0206
04181000 T  0210
04182000 T  0213
04183000 T  0213
04184000 T  0215
04184050 T  0216
04184100 T  0217
04184200 T  0221
04185000 T  0224
04186000 T  0225
04187000 T  0227
04188000 T  0227
04189000 T  0228
04190000 T  0228
04191000 T  0229
04192000 T  0229
04193000 T  0232
04194000 T  0232
04195000 T  0234
04196000 T  0234
04197000 T  0237
04198000 T  0237
04198100 T  0238
04199000 T  0241
04200000 T  0241
04201000 T  0243
04202000 T  0245
04203000 T  0246
04204000 T  0247
04205000 T  0248
04206000 T  0249
04207000 T  0249
04208000 T  0251
04208100 T  0251
04209000 T  0252
04210000 T  0257
04211000 T  0257

04212000 T  0258
04213000 T  0258
04214000 T  0258
04215000 T  0261
04216000 T  0262
04217000 T  0264
04218000 T  0264
04219000 T  0264
04220000 T  0267
04221000 T  0268
04222000 T  0270
```

```
                    IF T2 = LOGTYPE THEN EMITO(LNG) ELSE FLAG(53);                                  04223000 T  0270
                    BEGIN T ← LESS; GO TO RELATION END;                                             04224000 T  0274
                    BEGIN T ← LEQL; GO TO RELATION END;                                             04225000 T  0276
                    BEGIN T ← EQUL; GO TO RELATION END;                                             04226000 T  0277
                    BEGIN T ← GRTR; GO TO RELATION END;                                             04227000 T  0279
                    BEGIN T ← GEQL; GO TO RELATION END;                                             04228000 T  0281
                    BEGIN T ← NEQL;                                                                 04229000 T  0283
                       RELATION:                                                                    04230000 T  0283
                       IF CODE < 0 THEN FLAG(54) ELSE                                               04231000 T  0284
                       CASE CODE OF                                                                 04232000 T  0286
                       BEGIN   ;                                                                    04233000 T  0286
PRT(722) = *CASE STATEMENT DESCRIPTOR*
                          BEGIN                                                                     04234000 T  0287
                               EO(CHS); EP(9,STD); EO(XCH); EOL(9); EO(XCH); EP(0,XCH) ;            04235000 T  0287
                               EO(AD2) ;                                                            04236000 T  0292
                          END;                                                                      04238000 T  0293
                          FLAG(90);                                                                 04239000 T  0293
                          BEGIN EMITPAIR(0, XCH); EMITO(SB2) END;                                   04240000 T  0294
                          EMITO(SB2);                                                               04241000 T  0297
                          FLAG(90);                                                                 04242000 T  0298
                          FLAG(90);                                                                 04243000 T  0299
                          FLAG(90);                                                                 04244000 T  0300
                          IF T≠ EQUL AND T≠ NEQL THEN FLAG(54)                        %103-         04245000 P  0302
                          ELSE                                                        %103-         04245100 C  0304
                              BEGIN                                                   %103-         04245200 C  0305
                              EP(9,STD); EO(XCH); EOL(9); EO(T);                      %103-         04245300 C  0305
                              EP(9,STD ); EO(T); EOL(9);                              %103-         04245400 C  0308
                              T←(IF T=EQUL THEN LND FLSE LOR); CODE←0;                %103-         04245500 C  0311
                              END;                                                    %103-         04245600 C  0314
                       END RELATION CASE STATEMENT;                                                 04246000 T  0315

                                                                                  START OF SEGMENT **********  122
                                                                                   122 IS   10 LONG, NEXT SEG  121
                       IF CODE > 0 THEN                                                             04247000 T  0315
                       BEGIN EMITO(XCH); EMITO(DFL); EMITL(0) END;                                  04248000 T  0316
                       EMITO(T);                                                                    04249000 T  0319
                       OPTYPE[IT←IT-1] ← LOGTYPE;                                                   04250000 T  0320
                    END;                                                                            04251000 T  0322
                    IF CODE < 0 THEN BEGIN FLAG(55); IT ← IT-1 END ELSE                             04252000 T  0322
                    CASE CODE OF                                                                    04253000 T  0326
                    BEGIN                                                                           04254000 T  0326
PRT(723) = *CASE STATEMENT DESCRIPTOR*
                       BEGIN                                                                        04255000 T  0327
                          EMITO(ADD);                                                               04256000 T  0327
                          IF T1 = INTYPE AND T2 = INTYPE THEN IT ← IT-1 ELSE                        04257000 T  0328
                             OPTYPE[IT←IT-1] ← REALTYPE;                                            04258000 T  0331
                       END;                                                                         04259000 T  0334
                       BEGIN TM←AD2 ;                                                               04260000 T  0334
                 RPLUSD;  EP(9,STD); EO(XCH); EOL(9); EO(XCH); EP(0,XCH); EO(TM) ;                  04260010 T  0335
                 DTYP;    OPTYPE[IT←IT-1]←DOUBTYPE ;                                               04260020 T  0341
                       END ;                                                                        04260030 T  0343
                       BEGIN TM←ADD; GO RLESSC END ;                                                04261000 T  0343
                       BEGIN                                                                        04262000 T  0345
                          EMITPAIR(0, XCH);                                                         04263000 T  0345
                          EMITO(AD2);                                                               04264000 T  0346
                          IT ← IT-1;                                                                04265000 T  0347
                       END;                                                                         04266000 T  0348
                       BEGIN EMITO(AD2); IT ← IT-1 END;                                             04267000 T  0349
```

```
                        BEGIN TM←ADD; GO DLESSC END ;                                    04268000 T  0351
                        BEGIN EMITO(ADD); IT ← IT-1 END;                                 04269000 T  0353
                        BEGIN TM←ADD; GO CLESSD END ;                                    04270000 T  0355
                        BEGIN TM←ADD; GO CLESSC END ;                                    04271000 T  0357
                     END ADD CASE STATEMENT;                                             04272000 T  0359
                                                                                  START OF SEGMENT *********  123
                                                                                   123 IS   10 LONG, NEXT SEG  121
                     IF CODE < 0 THEN BEGIN FLAG(55); IT ← IT-1 END ELSE             04273000 T  0360
                     CASE CODE OF                                                    04274000 T  0363
                     BEGIN                                                           04275000 T  0364
PRT(724) = *CASE STATEMENT DESCRIPTOR*
                        BEGIN                                                        04276000 T  0364
                           EMITO(SUB);                                               04277000 T  0364
                           IF T1 = INTYPE AND T2 = INTYPE THEN IT ← IT-1 ELSE         04278000 T  0365
                              OPTYPE[IT←IT-1] ← REALTYPE;                            04279000 T  0369
                        END;                                                         04280000 T  0371
                        BEGIN EO(CHS); TM←AD2; GO RPLUSD END ;                       04281000 T  0372
                        BEGIN TM←SUB ;                                               04282000 T  0374
                  RLESSC: ES1(TM); GO DLESSC1 ;                                      04282010 T  0375
                        END ;                                                        04282030 T  0379
                        BEGIN                                                        04283000 T  0379
                           EMITPAIR(0, XCH);                                         04284000 T  0379
                           EMITO(SB2);                                               04285000 T  0380
                           IT ← IT-1;                                                04286000 T  0381
                        END;                                                         04287000 T  0382
                        BEGIN EMITO(SB2); IT ← IT-1 END;                             04288000 T  0383
                        BEGIN TM←SUB ;                                               04289000 T  0385
                  DLESSC: ES1(TM); EO(XCH); EO(DEL) ;                                04289005 T  0386
                  DLESSC1: EOL(9); IF TM=SUB THEN EO(CHS); GO CTIMESR2 ;             04289007 T  0391
                        END ;                                                        04289010 T  0394
                        BEGIN EMITO(SUB); IT ← IT-1 END;                             04290000 T  0394
                        BEGIN TM←SUB ;                                               04291000 T  0397
                  CLESSD: EO(XCH); EO(DEL); EO(TM) ;                                 04291010 T  0398
                  CTYP:    OPTYPE[IT←IT-1]←COMPTYPE ;                                04291015 T  0400
                        END ;                                                        04291020 T  0403
                        BEGIN TM←SUB ;                                               04292000 T  0403
                  CLESSC: ES1(TM); GO CTIMESR1 ;                                     04292010 T  0404
                        END ;                                                        04292020 T  0408
                     END SUBTRACT CASE STATEMENT;                                    04293000 T  0408
                                                                                  START OF SEGMENT *********  124
                                                                                   124 IS   10 LONG, NEXT SEG  121
                     BEGIN % HANDLE NEGATIVE NUMBERS CASE STATEMENT,                 04293100 T  0409
                     EXPV←-EXPV ;                                                    04293200 T  0409
                     IF T2 ≤ REALTYPE THEN EMITO(CHS) ELSE                           04294000 T  0410
                     IF T2 = LOGTYPE THEN FLAG(55) ELSE                              04295000 T  0412
                     IF T2 = DOUBTYPE THEN EMITO(CHS) ELSE                           04296000 T  0415
                     IF T2 = COMPTYPE THEN                                           04297000 T  0417
                     BEGIN                                                           04298000 T  0418
                        EMITO(CHS); EMITO(XCH);                                      04299000 T  0419
                        EMITO(CHS); EMITO(XCH);                                      04300000 T  0420
                     END ELSE FLAG(55);                                              04301000 T  0422
                     END OF NEG NUMBERS CASE STATEMENT ;                             04301100 T  0423
                     IF CODE < 0 THEN BEGIN FLAG(55); IT ← IT-1 END ELSE             04302000 T  0424
                     CASE CODE OF                                                    04303000 T  0427
                     BEGIN                                                           04304000 T  0428
PRT(725) = *CASE STATEMENT DESCRIPTOR*
                        BEGIN                                                        04305000 T  0428
```

```
                        EMITO(MUL);
                        IF T1 = INTYPE AND T2 = INTYPE THEN IT ← IT-1 ELSE
                           OPTYPE[IT←IT-1] ← REALTYPE;
                     END;
                     BEGIN TM←ML2; GO RPLUSD END ;
                     BEGIN ES2; GO DTIMESC END ;
                     BEGIN
                        EMITPAIR(0, XCH);
                        EMITO(ML2);
                        IT ← IT-1;
                     END;
                     BEGIN EMITO(ML2); IT ← IT-1 END;
                     BEGIN ES2; EO(XCH); EO(DEL) ;
               DTIMESC: EOL(9); EOL(17); EO(MUL); GO CTYP ;
                     END ;
                     BEGIN TM←MUL ;
               CTIMESR: EP(9,SND); EO(TM) ;
               CTIMESR1:EO(XCH); EOL(9); EO(TM) ;
               CTIMESR2:EO(XCH); GO CTYP ;
                     END ;
                     BEGIN TM←MUL; GO CDIVBYD END ;
                     MATH(2, 26, COMPTYPE);
                  END MULTIPLY CASE STATEMENT;


                  IF CODE < 0 THEN BEGIN FLAG(55); IT ← IT-1 END ELSE
                  CASE CODE OF
                  BEGIN
PRT(726) = *CASE STATEMENT DESCRIPTOR*
                     IF T1 = INTYPE AND T2 = INTYPE THEN
                     BEGIN EMITO(IDV); IT ← IT-1 END ELSE
                     BEGIN EMITO(DIU); OPTYPE[IT←IT-1] ← REALTYPE END;
                     BEGIN
                     EP(9,STD); EP(17,STD); EP(0,XCH); EOL(17); EOL(9); EO(DV2) ;
                     GO DTYP ;
                     END ;
                     MATH(1, 29, COMPTYPE);
                     BEGIN
                        EMITPAIR(0, XCH);
                        EMITO(DV2);
                        IT ← IT-1;
                     END;
                     BEGIN EMITO(DV2); IT ← IT-1 END;
                     MATH(2, 32, COMPTYPE);
                     BEGIN TM←DIU; GO CTIMESR END ;
                     BEGIN TM←DIU ;
               CDIVBYD: EO(XCH); EO(DEL); GO CTIMESR ;
                     END ;
                     MATH(2, 35, COMPTYPE);
                  END OF DIVIDE CASE STATEMENT;


                  IF CODE < 0 THEN BEGIN FLAG(55); IT ← IT-1 END ELSE
                  BEGIN
                  IF CODE = 0 AND T2 = INTYPE AND
                     CNSTSEENLAST THEN                                  %113-
                  BEGIN
```

```
04306000 T  0428
04307000 T  0429
04308000 T  0432
04309000 T  0435
04310000 T  0436
04311000 T  0437
04312000 T  0442
04313000 T  0442
04314000 T  0443
04315000 T  0444
04316000 T  0445
04317000 T  0445
04318000 T  0448
04318010 T  0453
04318020 T  0456
04319000 T  0457
04319010 T  0458
04319020 T  0459
04319030 T  0462
04319040 T  0464
04320000 T  0464
04321000 T  0466
04322000 T  0468
START OF SEGMENT **********  125
 125 IS   10 LONG, NEXT SEG  121
04323000 T  0469
04324000 T  0472
04325000 T  0473

04326000 T  0473
04327000 T  0475
04328000 T  0478
04329000 T  0482
04329010 T  0482
04329020 T  0487
04329030 T  0487
04330000 T  0488
04331000 T  0490
04332000 T  0490
04333000 T  0491
04334000 T  0491
04335000 T  0493
04336000 T  0493
04337000 T  0496
04338000 T  0497
04339000 T  0499
04339010 T  0500
04339020 T  0503
04340000 T  0503
04341000 T  0505
START OF SEGMENT **********  126
 126 IS   10 LONG, NEXT SEG  121
04342000 T  0506
04343000 T  0509
04344000 T  0510
04345000 P  0511
04346000 T  0512
```

```
                    IF T1 = INTYPE AND T2 = INTYPE THEN IT ← IT-1 ELSE                    04347000 T  0512
                    OPTYPE[IT←IT-1] ← REALTYPE;                                          04348000 T  0516
                    EXPV:=LINK;                                          %113-           04349000 P  0519
                    A:=1; ADR:=SAVEADR;                                  %113-           04350000 P  0519
                    WHILE EXPV  DIV 2 ≠ 0 DO                                             04351000 T  0521
                    BEGIN                                                                04352000 T  0523
                       EMITO(DUP);                                                       04353000 T  0523
                       IF BOOLEAN(EXPV) THEN BEGIN A←A+1; EMITO(DUP) END;                04354000 T  0524
                       EMITO(MUL);                                                       04355000 T  0527
                       EXPV ← EXPV DIV 2;                                                04356000 T  0528
                    END;                                                                 04357000 T  0529
                    IF EXPV = 0 THEN BEGIN EMITO(DEL); EMITL(1) END ELSE                 04358000 T  0529
                    WHILE A ← A-1 ≠ 0 DO EMITO(MUL);                                     04359000 T  0532
                 END ELSE                                                                04360000 T  0536
                 BEGIN                                                                   04361000 T  0536
                    EMITO(MKS);                                                          04362000 T  0537
                    EMITL(CODE);                                                         04363000 T  0537
                    EMITV(NEED(".XTOI ", INTRFUNID));                                    04364000 T  0538
                    CASE CODE OF                                                         04365000 T  0540
                    BEGIN                                                                04366000 T  0540
PRT(727) = *CASE STATEMENT DESCRIPTOR*
                    BEGIN EMITO(DEL); OPTYPE[IT←IT-1]←IF (T1=INTYPE AND T2=INTYPE)       04367000 T  0540
                                  THEN INTYPE ELSE REALTYPE END;                         04367500 T  0544
                    BEGIN EMITO(DEL); OPTYPE[IT←IT-1] ← DOUBTYPE END;                    04368000 T  0547
                    BEGIN EMITO(DEL); OPTYPE[IT←IT-1]←COMPTYPE END ;                     04369000 T  0550
                    BEGIN EMITO(DEL); IT ← IT-1 END;                                     04370000 T  0554
                    BEGIN EMITO(DEL); EMITO(DEL); IT ← IT-1 END;                         04371000 T  0556
                    BEGIN EMITO(DEL); EMITO(DEL); OPTYPE[IT←IT-1]←COMPTYPE END ;         04372000 T  0560
                    BEGIN EMITO(DEL); IT ← IT-1 END;                                     04373000 T  0564
                    BEGIN EMITO(DEL); EMITO(DEL); IT ← IT-1 END;                         04374000 T  0566
                       BEGIN EMITO(DEL); EMITO(DEL); IT←IT-1 END ;                       04375000 T  0570
                    END OF POWER CASE STATEMENT;                                         04376000 T  0573
                                                                                      START OF SEGMENT **********  127
                                                                                       127 IS   10 LONG, NEXT SEG  121
                 END;                                                                    04377000 T  0573
              END;                                                                       04378000 T  0573
                    END;                                                                 04379000 T  0574
                                                                                      START OF SEGMENT **********  128
                                                                                       128 IS   17 LONG, NEXT SEG  121
                 IP ← IP-1;                                                              04380000 T  0575
              END;                                                                       04381000 T  0576
              EXPRESLT ← EXPCLASS;                                                       04381100 T  0578
              STACK:                                                                     04382000 T  0578
              PR[IP←IP+1] ← PREC;                                                        04383000 T  0579
              OPST[IP] ← OP;                                                             04384000 T  0581
                                                                                         04385000 T  0582
              IF PREC > 0 AND PREC ≤ 4 THEN                                              04386000 T  0582
              BEGIN                                                                      04387000 T  0584
                 SCAN; LINK ← FNEXT;                                                     04388000 T  0584
                 IF NEXT = PLUS THEN GO TO LOOP;                                         04389000 T  0586
                 IF NEXT ≠ MINUS THEN GO TO NOSCAN;                                      04390000 T  0587
                 PREC ← 8; OP ← 12;                                                      04391000 T  0588
                 GO TO STACK;                                                            04392000 T  0590
              END;                                                                       04393000 T  0590
              GO TO LOOP;                                                                04394000 T  0590
              XIT: IF IP ≠ SAVIP THEN FLOG(56);                                          04395000 T  0591
              IP ← SAVIP-1;                                                              04396000 T  0593
```

```
                    EXPR ← OPTYPE[IT];                                                                  04397000 T  0594
                    IF OPTYPE[IT-1] ≠ 0 THEN FLOG(56);                                                  04398000 T  0595
                    IT ← SAVIT-1;                                                                       04399000 T  0598
                    EXPRESULT ← EXPRESLT;                                                               04400000 T  0599
                    EXPVALUE ← EXPV;                                                                    04401000 T  0600
                    EXPLINK ← EXPLNK;                                                                   04402000 T  0600
                IF DEBUGTOG THEN FLAGROUTINE(" EXPRE","SSION ",FALSE) ;                                 04403000 T  0601
                END EXPR;                                                                               04404000 T  0603
                                                                                                121 IS  610 LONG, NEXT SEG     6
                PROCEDURE FAULT (X);                                                                    04404050 T  0513
PRT(730) = FAULT
                    VALUE X;                                                                            04404100 T  0513
                    REAL X;                                                                             04404125 T  0513
                    BEGIN REAL LINK; LABEL XIT;                                                         04404150 T  0513
                                                                                              START OF SEGMENT **********  129
STACK(F+2) = LINK
                        SCAN; IF GLOBALNEXT ≠ LPAREN THEN BEGIN FLAG(106); GO XIT END;                  04404200 T  0000
                        SCAN; IF GLOBALNEXT ≠ ID THEN BEGIN FLAG(66); GO TO XIT END;                    04404250 T  0003
                        IF X = 1 THEN PDPRT[0,0] ← PDPRT[0,0] & 1[44:47:1] ELSE                         04404275 T  0006
                        PDPRT[0,0] ← PDPRT [0,0] & 1[43  :47:1];                                        04404300 T  0011
                        EMITOPDCLIT(41);   EMITO(DUP);                                                  04404325 T  0015
                        IF X = 1 THEN BEGIN EMITL(2); EMITO(XCH); EMITL(1) END                          04404350 T  0017
                                    ELSE EMITL(6);                                                      04404375 T  0020
                        EMITO(LND);                                                                     04404425 T  0022
                        IF X = 2 THEN EMITL(3);                                                         04404435 T  0022
                        EMITO(SUB);                                                                     04404450 T  0024
                        IF X = 2 THEN                                                                   04404475 T  0025
                          BEGIN EMITO(DUP); EMITL(3); EMITO(SSN) ;EMITO(EQUL); EMITL(2)                 04404500 T  0026
                               ;EMITO(BFC) ; EMITO(DEL);EMITL(2);                                       04404525 T  0030
                          END;                                                                          04404550 T  0032
                        LINK ← GET(GETSPACE(FNEXT)); EMITPAIR(LINK,ADDR,ISD);                           04404570 T  0032
                        IF X = 1 THEN EMITL(30) ELSE EMITL(25);                                         04404600 T  0036
                        EMITO(LND); EMITL(41);EMITO(STD);                                               04404625 T  0039
                        SCAN; IF GLOBALNEXT ≠ RPAREN THEN FLAG(108);                                    04404650 T  0041
                        SCAN;                                                                           04404660 T  0044
                    XIT;                                                                                04404675 T  0044
                    END FAULT;                                                                          04404700 T  0045
                                                                                                129 IS   48 LONG, NEXT SEG     6
                PROCEDURE SUBREF;                                                                       04405000 T  0513
PRT(731) = SUBREF
                BEGIN REAL LINK,INFC;                                                                   04406000 T  0513
                                                                                              START OF SEGMENT **********  130
STACK(F+2) = LINK
STACK(F+3) = INFC
                    REAL ACCIDENT;                                                                      04406010 T  0000
STACK(F+4) = ACCIDENT
                    LABEL XIT;                                                                          04406020 T  0000
                IF DEBUGTOG THEN FLAGROUTINE("   SUB","REF    ",TRUE ) ;                             04406025 T  0000
                IF TSSEDITOG THEN IF NAME="ZIP   " AND NOT DCINPUT THEN TSSED(NAME,3) ;                 04406030 T  0002
                   IF NAME = "EXIT  " THEN                                                              04407000 T  0006
                   BEGIN                                                                                04408000 T  0007
                      RETURNFOUND ← TRUE;                                                               04408100 T  0008
                      EMITL(1);                                                                         04409000 T  0009
                      EMITPAIR(16, STD);                                                                04410000 T  0010
                      EMITPAIR(10, KOM);                                                                04411000 T  0011
                      EMITPAIR( 5, KOM);                                                                04412000 T  0012
                      PUT(FNEXT+1, ".....");                                                            04413000 T  0013
```

```
   SCAN;                                                                                    04414000 T  0015
END ELSE IF NAME="ZIP    " AND NOT DCINPUT THEN                                           04415000 T  0015
BEGIN                                                                                       04415010 T  0023
   EMITO(MKS);                                                                              04415011 T  0023
   EMITL(0); EMITL(0); % DUMMY FILE AND FORMAT                                              04415020 T  0024
   EMITPAIR(-1,SSN);                                                                        04415021 T  0025
   EMITB(-1,FALSE); LADR1←LAX; ADJUST; DESCREQ←FALSE;                                       04415030 T  0027
 IF ADR ≥ 4085 THEN BEGIN ADR←ADR+1; SEGOVF END;                                            04415040 T  0030
   ACCIDENT←PRGDESCBLDR(0,0,ADR,[36:10]+1,NSEG);                                            04415050 T  0033
   EMITOPDCLIT(19);                                                                         04415070 T  0036
   EMITO(GFW);                                                                              04415080 T  0037
   LISTART ← ADR&NSEG[TOSEGNO]; ADJUST;SCAN;                                                04415090 T  0037
   IF GLOBALNEXT≠LPAREN THEN BEGIN FLAG(106);GO TO XIT END;                                 04415100 T  0040
   SCAN; IF GLOBALNEXT≠ID THEN BEGIN FLAG(66); GO TO XIT END;                               04415110 T  0045
   LINDX ← FNEXT; SCAN; XTA ← GET(LINDX+1);                                                 04415120 T  0048
   IF GLOBALNEXT≠RPAREN THEN BEGIN FLAG(108); GO TO XIT END;                                04415130 T  0051
   LINDX ← GETSPACE(LINDX);                                                                 04415140 T  0053
   IF T←(LINFA←GET(LINDX)).CLASS≠ARRAYID THEN                                               04415150 T  0054
            BEGIN FLAG(66); GO TO XIT END;                                                  04415160 T  0057
   IF XREF THEN ENTERX(XTA,0&LINFA[15:15:9]);                                               04415165 T  0059
   EMITPAIR(LADDR←LINFA.ADDR,LOD);                                                          04415170 T  0062
   IF BOOLEAN(LINFA.FORMAL) THEN                                                            04415180 T  0064
      BEGIN                                                                                 04415190 T  0064
      IF T ← GET(LINDX+2)<0 THEN EMITOPDCLIT(T,SIZE)                                        04415200 T  0065
      ELSE EMITNUM(T,SIZE); EMITOPDCLIT(LADDR-1); EMITO(CTF) END                            04415210 T  0068
   ELSE EMITNUM(GET(LINDX+2),BASENSIZE); EMITL(18); EMITO(STD);;                            04415220 T  0073
   EMITL(LINFA,CLASNSUB&0[44:47:1]); EMITL(19); EMITO(STD);                                 04415230 T  0077
   BRANCHLIT(LISTART,TRUE); EMITL(19); EMITO(STD);                                          04415240 T  0081
   EMITO(RTS); ADJUST;                                                                      04415250 T  0083
   EMITL(1); EMITO(CHS); EMITL(19); EMITO(STD);                                             04415260 T  0084
      EMITDESCLIT(19); EMITO(RTS); FIXB(LADR1); DESCREQ←FALSE;                              04415270 T  0087
      EMITPAIR(ACCIDENT,LOD); EMITOPDCLIT(7); EMITO(FTF);                                   04415280 T  0090
      EMITL(6);       %  EDITCODE 6 FOR ZIP                                                04415290 T  0093
      EMITV(NEED(".FTOUT",INTRFUNID));  SCAN                                             04415300 T  0094
    END ELSE IF NAME = "OVERFL" THEN FAULT(2)                                            04415310 T  0095
    ELSE IF NAME = "DVCHK " THEN FAULT(1)                                                04415320 T  0099
    ELSE                                                                                 04415330 T  0103
   BEGIN                                                                                 04416000 T  0104
   LINK ← NEED(NAME, SUBRID);                                                            04417000 T  0106
   IF XREF THEN ENTERX(XTA,0&GET(LINK)[15:15:5]);                                        04417100 T  0107
   EMITO(MKS);                                                                           04418000 T  0110
   SCAN;                                                                                 04419000 T  0111
   IF GLOBALNEXT = LPAREN THEN                                                           04420000 T  0112
      BEGIN PARAMETERS(LINK); SCAN END ELSE                                              04421000 T  0112
         IF NOT BOOLEAN((INFC←GET(LINK+2)),[1:1]) THEN                                   04422000 T  0114
      PUT(LINK+2,-INFC) ELSE                                                             04423000 T  0117
   IF INFC.NEXTRA ≠ 0 THEN                                                               04424000 T  0119
      BEGIN XTA ← GET(LINK+1); FLAG(28) END;                                             04425000 T  0121
   EMITV(LINK);                                                                          04426000 T  0124
                                                                                         04426500 T  0125
                                                                                         04426700 T  0125
   END;                                                                                  04427000 T  0125
    XIT;                                                                                 04427010 T  0125
IF DEBUGTOG THEN FLAGROUTINE("   SUB","REF    ",FALSE) ;                                04427025 T  0126
END SUBREF;                                                                              04428000 T  0128
                                                                                  130 IS  133 LONG, NEXT SEG    6
PROCEDURE DECLAREPARMS(FNEW); VALUE FNEW; REAL FNEW;                                     04429000 T  0513
```

```
PRT(732) = DECLAREPARMS
                BEGIN
                   REAL I, T, NLABELS, !NFA, !NFB, !NFC;                                                    04430000 T  0513
                                                                                                            04431000 T  0513
                                                                                                  START OF SEGMENT **********  131
STACK(F+2) = I
STACK(F+3) = T
STACK(F+4) = NLABELS
STACK(F+5) = !NFA
STACK(F+6) = !NFB
STACK(F+7) = !NFC
                   IF DEBUGTOG THEN FLAGROUTINE("DECLAR","EPARMS",TRUE ) ;                                  04431010 T  0000
                      !NFA ← GET(FNEW);                                                                     04432000 T  0002
                      IF !NFA.SEGNO ≠ 0 THEN BEGIN XTA ← NNEW; FLAG(25) END;                                04433000 T  0003
                      !NFA.SEGNO ← NSEG; PUT(FNEW,!NFA);                                                    04434000 T  0006
                      ENTRYLINK[ELX] ← 0 & FNEW[TOLINK] & NEXTSS[TOADDR];                                   04435000 T  0009
                      FOR I ← 1 STEP 1 UNTIL PARMS DO                                                       04436000 T  0012
                      BEGIN                                                                                 04437000 T  0016
                      EXTRAINFO[NEXTSS.IR,NEXTSS.IC] ← PARMLINK[I];                                         04438000 T  0016
                      NEXTSS ← NEXTSS+1;                                                                    04439000 T  0019
                      IF T ← PARMLINK[I] ≠ 0 THEN                                                           04440000 T  0020
                         BEGIN                                                                              04441000 T  0022
                            GETALL(T,!NFA,!NFB,!NFC);                                                       04442000 T  0022
                            IF BOOLEAN(!NFA   ,FORMAL) THEN                                                 04443000 T  0024
                            BEGIN                                                                           04443100 T  0024
                               IF !NFA.SEGNO = ELX THEN                                                     04444000 T  0025
                               BEGIN XTA ← !NFB   ; FLAG(26) END;                                           04445000 T  0026
                            END ELSE IF (!NFA < 0 AND !NFA.ADDR < 1024) OR BOOLEAN(!NFA.CE)                 04445100 T  0028
                                    THEN BEGIN XTA ← !NFB; FLAG(107) END;                                   04445200 T  0031
                            !NFA    ← !NFA    & 1[TOFORMAL] &  ELX[TOSEGNO];                                04446000 T  0034
                            !NFC   .BASE ← I;                                                               04447000 T  0037
                            PUT(T,!NFA); PUT(T+2,!NFC);                                                     04448000 T  0038
                         END ELSE NLABELS ← NLABELS+1;                                                      04449000 T  0040
                   END;                                                                                     04450000 T  0044
                   IF NLABELS > 0 THEN                                                                      04451000 T  0046
                   BEGIN ENTRYLINK[ELX  ].CLASS ← NLABELS;                                                  04452000 T  0047
                      IF LABELMOM=0 THEN BEGIN BUMPLOCALS; LABELMOM←LOCALS+1536 END;                        04453000 T  0050
                   END;                                                                                     04454000 T  0056
                   GETALL(FNEW,!NFA,!NFB,!NFC);                                                             04455000 T  0056
                   IF BOOLEAN(!NFC.[1:1]) THEN                                                              04456000 T  0058
                      BEGIN                                                                                 04457000 T  0058
                         IF !NFC.NEXTRA ≠ PARMS THEN                                                        04458000 T  0059
                            BEGIN XTA ← !NFB; FLOG(41);                                                     04459000 T  0060
                               PARMS ← !NFC.NEXTRA;                                                         04460000 T  0062
                            END;                                                                            04461000 T  0063
                         T ← !NFC.ADINFO;                                                                   04462000 T  0063
                         FOR I ← 1 STEP 1 UNTIL PARMS DO                                                    04463000 T  0065
                         IF NOT(PARMLINK[I] = 0 EQV                                                         04464000 T  0067
                            EXTRAINFO[(T+I-1).IR,(T+I-1).IC].CLASS = LABELID) THEN                          04465000 T  0068
                            BEGIN IF PARMLINK[I] = 0 THEN XTA ← "*      "                                   04466000 T  0073
                               ELSE XTA ←  GET(PARMLINK[I]+1);                                              04467000 T  0076
                               FLAG(40);                                                                    04468000 T  0080
                            END;                                                                            04469000 T  0080
                      END                                                                                   04470000 T  0083
                      ELSE                                                                                  04471000 T  0083
                      BEGIN                                                                                 04472000 T  0083
                         IF PARMS = 0 THEN !NFC ← -!NFC ELSE                                                04473000 T  0083
                         !NFC ← -(!NFC & PARMS[TONEXTRA]                                                    04474000 T  0085
```

```
                        & NEXTEXTRA[TOADINFO]);                                          04475000 T  0087
                     PUT(FNEW+2,INFC);                                                   04476000 T  0089
                     FOR I ← 1 STEP 1 UNTIL PARMS DO                                     04477000 T  0090
                     BEGIN                                                               04478000 T  0092
                        EXTRAINFO[NEXTEXTRA,IR,NEXTEXTRA,IC] ← 0 &                       04479000 T  0092
                           (IF PARMLINK[I] = 0 THEN LABELID ELSE 0)[TOCLASS];            04480000 T  0094
                        NEXTEXTRA ← NEXTEXTRA+1;                                         04481000 T  0098
                     END;                                                                04482000 T  0099
                  END;                                                                   04483000 T  0101
               IF ELX ← ELX+1 > MAXEL THEN BEGIN FLAG(128); ELX ← 0 END;                 04484000 T  0101
            IF DEBUGTOG THEN FLAGROUTINE("DECLAR","EPARMS",FALSE) ;                      04484010 T  0105
            END DECLAREPARMS;                                                            04485000 T  0107
                                                                                  131 IS  113 LONG, NEXT SEG    6
            PROCEDURE TOLIST(LEVEL);   REAL LEVEL;                                       04486000 T  0513
PRT(733) = TOLIST
            BEGIN ALPHA LADR2,T;                                                         04487000 T  0513
                                                                                START OF SEGMENT **********  132
STACK(F+2) = LADR2
STACK(F+3) = T
            BOOLEAN A;                                                                   04487050 T  0000
STACK(F+4) = A
            INTEGER INDX,I,BDLINK,NSUBS;                                                 04487100 T  0000
STACK(F+5) = INDX
STACK(F+6) = I
STACK(F+7) = BDLINK
STACK(F+10) = NSUBS
               LABEL ROUND,XIT,ERROR,LOOP,SCRAM;                                         04488000 T  0000
            INTEGER STREAM PROCEDURE CNTNAM(IDEN); VALUE IDEN;                           04488100 T  0000
PRT(734) = CNTNAM
            BEGIN  LABEL  XIT;                                                           04488200 T  0000
      SI := LOC IDEN; SI := SI +  3; TALLY := 1;                                         04488300 T  0000
      5(IF SC = " " THEN JUMP OUT TO XIT;SI := SI+1;TALLY := TALLY+1);                   04488400 T  0000
      XIT:  CNTNAM := TALLY;                                                             04488500 T  0002
END CNTNAM;                                                                              04488600 T  0003
                                                                                         04489000 T  0004
                                                                                         04490000 T  0004
IF DEBUGTOG THEN FLAGROUTINE("   IOL","IST   ",TRUE ) ;                                 04491000 T  0004
ROUND:  DESCREQ ← TRUE;                                                                  04492000 T  0007
        LOCALNAME := FALSE;                                                              04492100 T  0007
IF GLOBALNEXT = SEMI THEN GO TO XIT;                                                     04493000 T  0009
IF GLOBALNEXT = STAR THEN                                                                04493100 T  0010
    BEGIN IF NOT NAMEDESC THEN                                                           04493150 T  0011
             TV := ENTER(0&LISTSID[TOCLASS],LISTID:=LISTID+1);                           04493200 T  0013
          LOCALNAME := TRUE; NAMEDESC := TRUE; SCAN;                                     04493300 T  0017
    END;                                                                                 04493350 T  0021
IF GLOBALNEXT = ID THEN                                                                  04494000 T  0021
BEGIN LINDX ← FNEXT;                                                                     04495000 T  0021
      SCAN; XTA ← GET(LINDX+1);                                                          04496000 T  0023
      IF GLOBALNEXT = EQUAL THEN        %RETURN TO CALLER                                04497000 T  0025
      BEGIN IF (CLINFA←GET(GETSPACE(LINDX))).CLASS ≠ VARID THEN FLAG(50);                04498000 T  0026
            SCRAM: IF (LEVEL ← LEVEL-1) < 0 THEN FLOG(97);                               04498100 T  0030
            GO TO XIT;                                                                   04498200 T  0034
      END;                                                                               04499000 T  0037
                                                                                         04500000 T  0037
      IF DATASTMTFLAG AND SPLINK ≥ 0 THEN        %DECLARE OWN                            04500100 T  0037
      BEGIN                                                                              04500200 T  0038
         IF BOOLEAN(GET(LINDX),FORMAL) THEN FLAG(147);                                   04500300 T  0039
```

```
   IF SPLINK>1 THEN                                                    04500310 T  0041
        IF GFT(LINDX).ADDR>1023 THEN       FLAG(174);                 04500320 T  0042
   LINDX ← GFTSPACE(-LINDX);                                           04500400 T  0046
   IF BOOLEAN(GET(LINDX).EQ) THEN FLAG(168);                           04500420 T  0047
END ELSE LINDX ← GETSPACE(LINDX);                                      04500500 T  0050
IF T ← (LINFA←GET(LINDX)).CLASS > VARID THEN FLAG(50);                 04500600 T  0051
IF XREF THEN ENTERX(XTA,C2&LINFA[15:15:9]);                            04500655 T  0055
IF GLOBALNAME OR LOCALNAME THEN                                        04500700 T  0058
   IF NAMEIND:= NAMEIND+1 GTR LSTMAX THEN FLOG(161)                    04500725 T  0060
   ELSE NAMLIST[NAMEIND] := XTA & CNTNAM(XTA)[9:45:3];                 04500750 T  0063
IF T = ARRAYID THEN                                                    04501000 T  0067
  IF GLOBALNEXT ≠ LPAREN THEN                                          04502000 T  0068
  BEGIN IF SPLINK ≠ 1 THEN                                             04503000 T  0069
        BEGIN                                                          04503004 T  0070
        EMITL(0);                                                      04503008 T  0071
        EMITPAIR(LADDR ← LINFA,ADDR,LOD);                              04503010 T  0072
        EMITO(FTC);                                                    04503020 T  0074
        EMITDESCLIT(2);                                                04503030 T  0074
        EMITO(INX);                                                    04503040 T  0075
        EMITO(LOD);                                                    04503050 T  0076
        END ELSE EMITPAIR(LADDR←LINFA,ADDR,LOD);                       04503055 T  0077
      NSUBS := (T := GET (LINDX+2)).NEXTRA;                            04503100 T  0079
      IF GLOBALNAME OR LOCALNAME THEN                                  04503125 T  0082
      BEGIN                                                            04503150 T  0084
     IF NSUBS GTR SAVESUBS THEN SAVESUBS := NSUBS;                     04503160 T  0084
     IF NSUBS GTR NAMLIST[0] THEN NAMLIST[0] := NSUBS;                 04503170 T  0086
           NAMLIST[NAMEIND].[1:8] := NSUBS;                            04503175 T  0089
      INDX := -1;                                                      04503180 T  0091
      INFA := GET(NEED(".SUBAR",BLOCKID)).ADDR;                        04503190 T  0092
      BDLINK := T.ADINFO+1;                                            04503200 T  0095
      END;                                                             04503225 T  0097
      IF BOOLEAN (LINFA.FORMAL) THEN                                   04504000 T  0097
      BEGIN                                                            04505000 T  0097
           IF T LSS 0 THEN EMITOPDCLIT(T,SIZE)                         04505200 T  0098
           ELSE EMITNUM(T,SIZE);                                       04506000 T  0099
           EMITOPDCLIT(LADDR - 1);                                     04507000 T  0104
           EMITO(CTF);                                                 04508000 T  0105
      END ELSE EMITNUM(T,BASENSIZE);                                   04509000 T  0106
      IF GLOBALNAME OR LOCALNAME THEN                                  04509050 T  0108
        FOR I := 1 STEP 1 UNTIL NSUBS DO                               04509100 T  0109
        BEGIN IF T := EXTRAINFO[(BDLINK:=BDLINK-1),IR,                 04509150 T  0111
              BDLINK,IC] LSS 0 THEN EMITOPDCLIT(T)                     04509200 T  0113
              ELSE EMITNUM(T);                                         04509250 T  0116
              EMITNUM(INDX := INDX+1);                                 04509300 T  0117
              EMITDESCLIT(INFA);                                       04509350 T  0119
              EMITO(STD);                                              04509400 T  0120
        END;                                                           04509450 T  0121
      EMITL(18);  EMITO(STD);                                          04510000 T  0123
  END ELSE                                                             04511000 T  0124
      BEGIN SCAN;                                                      04512000 T  0124
           A ←(IF GLOBALNAME OR LOCALNAME                              04513000 T  0125
           THEN SUBSCRIPTS(LINDX,4) ELSE SUBSCRIPTS(LINDX,2));         04513100 T  0126
           SCAN;                                                       04514000 T  0131
      END                                                              04515000 T  0131
ELSE EMITN(LINDX);                                                     04516000 T  0131
IF GLOBALNAME OR LOCALNAME THEN                                        04516100 T  0132
BEGIN EMITOPDCLIT(18); EMITNUM(NAMEIND);                               04516200 T  0134
```

```
            EMITD(43,DIA); EMITD(3,DIB); EMITD(15,TRB);                    04516300 T  0136
            EMITL(18); EMITO(STD);                                         04516350 T  0139
        END;                                                               04516400 T  0141
        EMITL(LINFA,CLASNSUB&0[44:47:1]);                                  04517000 T  0141
        EMITL(20);  EMITO(STD);                                            04518000 T  0143
      IF ADR > 4083 THEN                                                   04518100 T  0144
         BEGIN ADR←ADR+1;  SEGOVF END ;                                    04518200 T  0145
        BRANCHLIT(LISTART,TRUE);                                           04519000 T  0147
        EMITL(19);  EMITO(STD);                                            04520000 T  0148
        EMITO(RTS); ADJUST;                                                04521000 T  0150
        GO TO LOOP;                                                        04522000 T  0151
END;                                                                       04523000 T  0153
IF GLOBALNEXT = LPAREN THEN    %  RECURSE ON (                             04524000 T  0153
BEGIN EMITB(-1,FALSE);                                                     04525000 T  0153
        ADJUST;                                                            04526000 T  0155
        LADR2 ← (ADR + 1)&LAX[TOADDR]&NSEG[TOSEGNO];                       04527000 T  0156
        SCAN;  LEVEL ← LEVEL + 1;                                          04528000 T  0159
        IOLIST(LEVEL);                                                     04529000 T  0161
        IF GLOBALNEXT ≠ EQUAL THEN  %  PHONY IMP DO                        04530000 T  0162
        BEGIN BRANCHES[T ← LADR2.ADDR] ← BRANCHX;                          04531000 T  0163
              BRANCHX ← T;                                                 04532000 T  0165
              IF GLOBALNEXT ≠ RPAREN THEN GO TO ERROR;                     04533000 T  0166
              SCAN;  GO TO LOOP;                                           04534000 T  0167
        END;                                                               04535000 T  0168
        IF XREF THEN ENTERX(GET(LINDX+1),1&LINFA[15:15:9]);                04535500 T  0168
        IF LINFA.SUBCLASS > REALTYPE THEN                                  04536000 T  0172
        BEGIN XTA ← GET(LINDX + 1);                                        04537000 T  0173
              FLAG(84);                                                    04538000 T  0176
        END;                                                               04539000 T  0176
        EMITB(-1,FALSE);                                                   04540000 T  0176
        LADR3 ← LAX;                                                       04541000 T  0178
        FIXB(LADR2,ADDR);                                                  04542000 T  0178
        DESCREQ ← FALSE;                                                   04543000 T  0180
        SCAN;  IF EXPR(TRUE) > REALTYPE THEN FLAG(102);  % INITIAL VALUE   04544000 T  0180
        EMITN(LINDX);      EMITO(STD);                                     04545000 T  0183
        EMITB(LADR2,FALSE);                                                04546000 T  0185
        IF GLOBALNEXT ≠ COMMA THEN GO TO ERROR;                            04547000 T  0186
        ADJUST;                                                            04548000 T  0187
        LADR4 ← (ADR + 1)&NSEG[TOSEGNO];                                   04549000 T  0188
        SCAN;  IF EXPR(TRUE) > REALTYPE THEN FLAG(102) ELSE EMITO(GRTR);   04550000 T  0190
        EMITB(LADR2,TRUE);                                                 04551000 T  0194
        EMITB(-1,FALSE);                                                   04552000 T  0195
        LADR5 ← LAX;                                                       04553000 T  0196
        FIXB(LADR3);                                                       04554000 T  0197
        IF GLOBALNEXT ≠ COMMA THEN EMITL(1)                                04555000 T  0198
        ELSE BEGIN SCAN; IF EXPR(TRUE) > REALTYPE THEN FLAG(102); END;     04556000 T  0199
        EMITV(LINDX);  EMITO(ADD);                                         04557000 T  0203
        EMITN(LINDX);      EMITO(SND);                                     04558000 T  0205
        EMITB(LADR4,FALSE);                                                04559000 T  0206
        FIXB(LADR5);                                                       04560000 T  0207
        IF GLOBALNEXT = RPAREN THEN SCAN ELSE GO TO ERROR;                 04561000 T  0208
LOOP:  IF GLOBALNEXT = SEMI OR GLOBALNEXT = SLASH THEN GO TO XIT;          04562000 T  0210
           IF GLOBALNEXT = RPAREN THEN GO TO SCRAM;                        04563000 T  0213
             IF GLOBALNEXT = COMMA THEN                                    04564000 T  0214
             BEGIN SCAN;                                                   04565000 T  0215
                   IF GLOBALNEXT = SEMI THEN GO TO ERROR;                  04566000 T  0216
                   GO TO ROUND;                                            04567000 T  0217
```

```
                    END;                                                                    04568000 T  0218
                ERROR: XTA ← NAME;                                                          04569000 T  0218
                       FLAG(94);                                                            04570000 T  0219
                       IF GLOBALNEXT = SEMI THEN GO TO XIT;                                 04571000 T  0220
                       SCAN;                                                                04572000 T  0221
                       IF GLOBALNEXT = ID THEN GO TO ROUND;                                 04573000 T  0222
                       ERRORTOG ← TRUE; GO TO XIT;                                          04574000 T  0223
                END;                                                                        04575000 T  0224
                IF GLOBALNEXT = RPAREN THEN GO TO SCRAM ELSE                                04576000 T  0224
                  IF GLOBALNEXT ≠ SLASH THEN GO TO ERROR;                                   04577000 T  0225
                XIT: IF DEBUGTOG THEN FLAGROUTINE("   IOL","IST    ",FALSE) ;              04578000 T  0227
                END IOLIST;                                                                 04579000 T  0230
                                                                                  132 IS  236 LONG, NEXT SEG    6
                INTEGER PROCEDURE FILECHECK(FILENAME,FILETYPE);                             04580000 T  0513
PRT(735) = FILECHECK
                VALUE FILENAME,FILETYPE;   ALPHA FILENAME;   INTEGER FILETYPE;              04581000 T  0513
                BEGIN COMMENT  THIS PROCEDURE RETURNS THE PRT CELL ALLOCATED TO             04582000 T  0513
                THE FILE FILENAME... A CELL IS CREATED IF NONE EXISTS;                      04583000 T  0513
                IF DEBUGTOG THEN FLAGROUTINE(" FILEC","HECK  ",TRUE ) ;                     04583010 T  0513
                EMITL(IF NOTOPIO THEN 2 ELSE 5);   %  FOR IO DESCRIPTOR                    04584000 T  0516
                IF T ← GLOBALSEARCH(FILENAME) = 0 THEN   %  FILE UNDECLARED                04585000 T  0518
                BEGIN  MAXFILES ← MAXFILES + 1;                                             04586000 T  0520
                BUMPPRT;                                                                    04586500 T  0522
                I ← GLOBALENTER(=0&(FILECHECK←PRTS)[TOADDR]                                 04587000 T  0526
                                &FILEID[TOCLASS],FILENAME)+2;                               04588000 T  0528
                INFO[I,1,1,1C], LINK ← FILETYPE;                                            04589000 T  0530
                END ELSE    %  FILE ALREADY EXISTS                                         04590000 T  0535
                FILECHECK ←  GET(T).ADDR;                                                   04591000 T  0535
                IF DEBUGTOG THEN FLAGROUTINE(" FILEC","HECK  ",FALSE) ;                     04591010 T  0539
                END FILECHECK;                                                              04592000 T  0541
                PROCEDURE INLINEFILE;                                                       04593000 T  0546
PRT(736) = INLINEFILE
                BEGIN COMMENT THIS PROCEDURE GENERATES THE CODE TO BRING UP THE FILE...     04594000 T  0546
                  IF THE FILE IS AN INTEGER THEN FILECHECK IS CALLED, IF THE FILE           04595000 T  0546
                  IS NOT AN INTEGER THEN IN-LINE CODE IS GENERATED FOR OBJECT TIME          04596000 T  0546
                  ANALYSIS;                                                                 04597000 T  0546
                REAL TEST;                                                                  04598000 T  0546
                                                                        START OF SEGMENT **********  133
STACK(F+2) = TEST
                COMMENT IF LAST INSTRUCTION WAS A LIT CALL THEN WE HAVE SEEN REFERENCE      04599000 T  0000
                TO AN INTEGER FILE ID;                                                      04600000 T  0000
                IF DEBUGTOG THEN FLAGROUTINE(" INLIN","EFILE ",TRUE ) ;                     04600010 T  0000
                TEST←ADR ;                                                                  04601000 T  0002
                IF EXPR(TRUE)>REALTYPE THEN FLAG(102)                                       04602000 T  0002
                ELSE IF EXPRESULT=NUMCLASS THEN                                             04602500 T  0004
                        BEGIN XTA←NNEW ;                                                    04603000 T  0008
                        IF EXPVALUE≥1.0@5 OR EXPVALUE≤0.5 THEN FLAG(33)                     04603500 T  0010
                        ELSE BEGIN                                                          04604000 T  0012
                             IF ADR<TEST THEN %  EXPR GOT SEGOVFL--ASSUME NO WRAPAROUND     04604010 T  0016
                                  EMITO(DEL)                                                04604020 T  0016
                             ELSE ADR←TEST;                                                 04604025 T  0017
                             TEST←SPCLBIN2DEC(C1←EXPVALUE);                                 04604030 T  0019
                             IF XREF THEN ENTERX(TEST&1[TOCE],0&FILEID[TOCLASS]) ;          04604040 T  0021
                             EMITDESCLIT(FILECHECK(0&","[12:42:6]&TEST[18:12:30],           04604050 T  0025
                                         IF DCINPUT THEN 12 ELSE 2)) ;                      04604060 T  0028
                             END ;                                                          04604070 T  0031
                        END ELSE                                                            04604080 T  0031
```

```
                COMMENT  NOT INTEGER FILE... GENERATE OBJECT TIME CODE;                                  04605000 T  0031
                     BEGIN EMITPAIR(17,ISN) ;                                                           04606000 T  0031
                      IF FILEARRAYPRT=0 THEN BEGIN BUMPPRT;FILEARRAYPRT←PRTS END;                       04607000 T  0032
                      EMITOPDCLIT(FILEARRAYPRT);                                                        04608000 T  0038
                      EMITO(DUP);   EMITL(0);   EMITO(EQUL);                                            04609000 T  0039
                        COMMENT NEED 7 CONSECUTIVE SYLLABLES;                                           04610000 T  0041
                     IF ADR ≥ 4082 THEN                                                                 04611000 T  0041
                      BEGIN ADR ← ADR + 1;                                                              04612000 T  0042
                            SEGOVF;                                                                     04613000 T  0043
                      END;                                                                              04614000 T  0044
                      EMITL(3+FILEARRAYPRT.[38:1]);%CONSIDER POSSIBLE XRT                               04615000 T  0044
                      EMITO(BFC);                    % BRANCH AROUND TERMINATION CODE                  04615100 T  0046
                      EMITL(1);   EMITO(CHS);   EMITOPDCLIT(FILEARRAYPRT);  %  INV INDEX              04616000 T  0046
                      EMITO(LOD);   EMITL(IF NOTOPIO THEN 2 ELSE 5);   EMITO(CDC);                      04617000 T  0049
                END;                                                                                    04618000 T  0053
                IF DEBUGTOG THEN FLAGROUTINE(" INLIN","EFILE ",FALSE) ;                                 04618010 T  0053
                END INLINEFILE;                                                                         04619000 T  0055
                                                                                    133 IS   61 LONG, NEXT SEG    6
                PROCEDURE FILECONTROL(N); VALUE N; REAL N;                                              04619010 T  0546
PRT(737) = FILECONTROL
                BEGIN    COMMENT COMPILES ALL CALLS ON ALGOL FILE CONTROL;                              04619020 T  0546
                         EODS←TRUE ;                                                                    04619025 T  0546
                         EXECUTABLE ;                                                                   04619027 T  0547
                         SCAN; EMITO(MKS);                                                              04619030 T  0548
                         EMITL(N); EMITL(0);                                                            04619040 T  0549
                         NOTOPIO ← TRUE;                                                                04619050 T  0551
                         INLINEFILE ;                                                                   04619060 T  0552
                         EMITL(4); EMITOPDCLIT(14);                                                     04619070 T  0553
                         NOTOPIO ← FALSE;                                                               04619080 T  0554
                END FILECONTROL;                                                                        04619090 T  0556
                PROCEDURE FORMATER;                                                                     04620000 T  0556
PRT(740) = FORMATER
                BEGIN                                                                                   04621000 T  0556
                COMMENT FORMATER COMPILES A PSEUDO CODE USED BY THE OBJECT TIME                         04621500 T  0556
                        FORMATING ROUTINES TO PRODUCE DESIRED I/O. USUALLY, THERE IS                    04622000 T  0556
                        ONE WORD OF PSEUDO CODE PRODUCED FOR EACH EDITING PHRASE. IN                    04622500 T  0556
                        ADDITION ONE WORD IS PRODUCED FOR EACH RIGHT PARENTHESIS AND                    04623000 T  0556
                        SLASH GROUP.                                                                    04623500 T  0556
                        THE 47TH BIT OF THE FIRST WORD IS SET IF THERE ARE LIST                         04624000 T  0556
                        ELEMENT PHRASES IN THE FORMAT STATEMENT.                                        04624500 T  0556
                        THERE ARE FOUR GROUPS OF PHRASES:                                               04625000 T  0556
                        GROUP 1 = F, E, D, G              = REPEAT WIDTH DECIMAL                        04625500 T  0556
                        GROUP 2 = I, J, A, O, L, T, C,    = REPEAT WIDTH                                04626000 T  0556
                                  X, P                                                                  04626100 T  0556
                        GROUP 3 = H                       = REPEAT SPECIAL PURPOSE                      04626500 T  0556
                        GROUP 4 = ), /                    = CONTROL REPEAT LINK                         04627000 T  0556
                        WORD FOR GROUP 1:                                                               04627500 T  0556
                             CODE    =  [ 1: 5]                                                         04628000 T  0556
                             REPEAT  =  [ 6:12]                                                         04628500 T  0556
                             WIDTH   =  [18:12]                                                         04629000 T  0556
                             DEDIMAL =  [30:12]                                                         04629500 T  0556
                        WORD FOR GROUP 2:                                                               04630000 T  0556
                             CODE    =  [ 1: 5]                                                         04630500 T  0556
                             REPEAT  =  [ 6:12]                                                         04631000 T  0556
                             WIDTH   =  [18:12]                                                         04631500 T  0556
                             SIGN    = [41: 1]                                                          04631600 T  0556
                             FOR P, X AND T REPEAT = 1                                                  04631625 T  0556
```

```
                P: SIGN = SIGN OF SCALE                                                 04631650 T  0556
           WORD FOR GROUP 3:                                                            04632000 T  0556
                CODE    =  [ 1: 5]                                                      04632500 T  0556
                REPEAT  =  [ 6:12]                                                      04633000 T  0556
                H: STRING STARTS AT BIT 18 AND CONTINUES TO END OF STRING               04635500 T  0556
                   REPEAT = NUMBER OF STRING CHARACTERS                                 04635600 T  0556
           WORD FOR GROUP 4:                                                            04636000 T  0556
                CODE    =  [ 1: 5]                                                      04636500 T  0556
                REPEAT  =  [ 6:12]                                                      04637000 T  0556
                LINK    =  [18:12]                                                      04637500 T  0556
                DECIMAL =  [30:12]                                                      04638000 T  0556
                CNTROL  =  [47: 1]                                                      04638500 T  0556
                ): LINK NUMBER OF WORD - 1 BACK TO 1ST PHRASE AFTER                     04639000 T  0556
                      LEFT PAREN                                                        04639500 T  0556
                   REPEAT = REPEAT OF ASSOCIATED LEFT PAREN                             04639600 T  0556
                   OUTER MOST RIGHT PAREN                                               04640000 T  0556
                     A. DECIMAL ≠ 0                                                     04640500 T  0556
                     B. LINKS TO LEFT PAREN OF LAST PREVIOUS RIGHT PAREN                04641000 T  0556
                /: REPEAT = NUMBER OF SLASHES.                                          04641500 T  0556
           ;                                                                            04642000 T  0556
DEFINE     APHASE =  6 #,                                                               04642500 T  0556
                                                                                START OF SEGMENT **********  134
           VPHRASE = 30 #,                                                              04642510 T  0000
           LPPHRASE = 29 #,                                                             04642520 T  0000
           CPHASE = 15 #,                                                               04642600 T  0000
           DPHASE = 14 #,                                                               04643000 T  0000
           EPHASE = 13 #,                                                               04643500 T  0000
           FPHASE = 12 #,                                                               04644000 T  0000
           GPHASE = 11 #,                                                               04644500 T  0000
           HPHASE =  2 #,                                                               04645000 T  0000
           IPHASE = 10 #,                                                               04645500 T  0000
           JPHASE =  9 #,                                                               04646000 T  0000
           LPHASE =  8 #,                                                               04646500 T  0000
           OPHASE =  7 #,                                                               04647000 T  0000
           PPHASE =  3 #,                                                               04647500 T  0000
           TPHASE =  5 #,                                                               04648000 T  0000
           XPHASE =  4 #,                                                               04648500 T  0000
           RTPARN =  0 #,                                                               04649000 T  0000
           SLASH  =  1 #,                                                               04649500 T  0000
           TOCODE    =  1:43: 5 #,                                                      04650000 T  0000
           TOREPEAT  =  6:36:12 #,                                                      04650500 T  0000
           TOWIDTH   = 18:36:12 #,                                                      04651000 T  0000
           TOLINK    = 18:36:12 #,                                                      04651500 T  0000
           TODECIMAL = 30:36:12 #,                                                      04652000 T  0000
           TOCNTRL   = 47:47: 1 #,                                                      04652500 T  0000
           TONUM     = 42:43: 5 #,                                                      04653000 T  0000
           TOSIGN    = 41:47: 1 #,                                                      04653500 T  0000
           WSA       = LSTT #;                                                          04654000 T  0000
      REAL J, T;                                                                        04655000 T  0000
STACK(F+2) = J
STACK(F+3) = T
      LABEL KK,SL,FL ;                                                                  04655100 T  0000
      FORMAT FM(///"PHRASE#     CODE   REPEAT    WIDTH  DECIMAL",X2,                    04655400 T  0000
                                                                                START OF SEGMENT **********  135
PRT(741) = FM
           ".[41:1]  .[42:4]  .[42:5]  .[44:1]  .[45:1]",X2,                            04655410 T  0000
           ".[46:1],  .[46:2]  .[47:1]"/) ;                                             04655420 T  0000
```

```
                    INTEGER D,I,  CODE,REPEAT,WIDTH,DECIMAL,TOTAL,SLCNT,SAVLASTLP,          135 IS   29 LONG, NEXT SEG  134
                                                                                                04655500 T  0000
STACK(F+4) = D
STACK(F+5) = I
STACK(F+6) = CODE
STACK(F+7) = REPEAT
STACK(F+10) = WIDTH
STACK(F+11) = DECIMAL
STACK(F+12) = TOTAL
STACK(F+13) = SLCNT
STACK(F+14) = SAVLASTLP
                        PARENCT,SAVTOTAL;                                                   04656000 T  0000
STACK(F+15) = PARENCT
STACK(F+16) = SAVTOTAL
                      BOOLEAN VRB,ASK ;                                                     04656100 T  0000
STACK(F+17) = VRB
STACK(F+20) = ASK
                      BOOLEAN LISTEL, HF, ZF, QF, PF, MINUSP, FIELD, STRINGF;               04656500 T  0000
STACK(F+21) = LISTEL
STACK(F+22) = HF
STACK(F+23) = ZF
STACK(F+24) = QF
STACK(F+25) = PF
STACK(F+26) = MINUSP
STACK(F+27) = FIELD
STACK(F+30) = STRINGF
                      BOOLEAN COMMAS, DOLLARS, PLUSP, STR;                                  04656600 T  0000
STACK(F+31) = COMMAS
STACK(F+32) = DOLLARS
STACK(F+33) = PLUSP
STACK(F+34) = STR
                    LABEL ROUND,NOPLACE,NUM,NUL,LP,RP,ENDER,NOEND,FALL,SEMIC,NUL1;          04657000 T  0000
                    LABEL NEWWD, ROUND1;                                                    04657100 T  0000
                    ALPHA STREAM PROCEDURE GETCHAR(NCRV, T); VALUE NCRV;                    04657500 T  0000
PRT(742) = GETCHAR
                    BEGIN SI ← NCRV; DI ← T; DI ← DI+7; DS ← CHR; GETCHAR ← SI;             04658000 T  0000
                    END GETCHAR;                                                            04658500 T  0001
                    BOOLEAN PROCEDURE CONTINUE;                                             04659000 T  0002
PRT(743) = CONTINUE
                    BEGIN                                                                   04659500 T  0002
                    LABEL LOOP;                                                             04660000 T  0002
                                                                          START OF SEGMENT **********  136
                    BOOLEAN STREAM PROCEDURE CONTIN(CD);                                    04660500 T  0000
PRT(744) = CONTIN
                    BEGIN SI←CD; IF SC ≠ "C" THEN IF SC ≠ "$" THEN BEGIN SI←SI+5;           04661000 T  0000
                          IF SC ≠ " " THEN IF SC ≠ "0" THEN BEGIN TALLY←1;                  04661500 T  0002
                          CONTIN ← TALLY END END                                            04662000 T  0004
                    END OF CONTIN;                                                          04662500 T  0005
                    BOOLEAN STREAM PROCEDURE COMNT(CD,T); VALUE T;                          04663000 T  0006
PRT(745) = COMNT
                    BEGIN LABEL L;                                                          04663500 T  0006
                       SI←CD; IF SC = "C" THEN BEGIN T(SI←SI+1; IF SC="-" THEN               04664000 T  0006
                       TALLY←1 ELSE JUMP OUT TO L); TALLY←1; L: END; COMNT←TALLY;            04664500 T  0009
                    END COMNT;                                                              04665000 T  0011
                    BOOLEAN STREAM PROCEDURE DCCONTIN(CD);                                  04665500 T  0012
PRT(746) = DCCONTIN
                    BEGIN SI←CD; IF SC = "-" THEN TALLY ← 1; DCCONTIN←TALLY;                 04666000 T  0012
```

```
                END DCCONTIN;
                LOOP:   IF NOT(CONTINUE ←                                                                  04666500 T  0014
                        IF DCINPUT AND NOT TSSEDITOG OR FREEFTOG THEN                                      04667000 T  0015
                             IF NEXTCARD<4 THEN DCCONTIN(CB)                                               04667500 T  0016
                             ELSE IF NEXTCARD=7 THEN DCCONTIN(DB) ELSE CONTIN(TB)                          04667600 T  0019
                        ELSE IF NEXTCARD<4 THEN CONTIN(CB)                                                 04668000 T  0021
                             ELSE IF NEXTCARD=7 THEN CONTIN(DB) ELSE CONTIN(TB))                           04668500 T  0026
                        THEN IF(IF NEXTCARD < 4 THEN                                                       04668600 T  0030
                               COMNT(CB,DCINPUT AND NOT TSSEDITOG OR FREEFTOG)                             04669000 T  0035
                               ELSE IF NEXTCARD = 7 THEN                                                   04669500 T  0038
                               COMNT(DB,DCINPUT AND NOT TSSEDITOG OR FREEFTOG)                             04670000 T  0041
                        ELSE COMNT(TB,0) AND NEXTCARD≠6) THEN                                              04670100 T  0044
                        BEGIN                                                                              04670500 T  0048
                          IF READACARD THEN IF LISTOG THEN PRINTCARD;                                      04671000 T  0052
                          GO TO LOOP;                                                                      04671500 T  0053
                        END;                                                                               04672000 T  0055
                END CONTINUE;                                                                              04672500 T  0056
                                                                                                           04673000 T  0056
                STREAM PROCEDURE STORECHAR(ARY,POSIT,CHAR);VALUE POSIT;                          136 IS   59 LONG, NEXT SEG  134
PRT(747) = STORECHAR                                                                                       04673500 T  0002
                BEGIN SI ←CHAR; SI ← SI + 7;                                                               04674000 T  0002
                      DI ← ARY; DI ← DI + POSIT;                                                           04674500 T  0003
                      DS ← CHR;                                                                            04675000 T  0004
                END STORECHAR;                                                                             04675500 T  0004
                ALPHA STREAM PROCEDURE BACKNCR(NCRV); VALUE NCRV;                                          04676000 T  0004
PRT(750) = BACKNCR                                                                                         04676500 T  0004
                BEGIN SI←NCRV; SI ← SI-1; BACKNCR ← SI; END BACKNCR;                                       04677000 T  0006
                COMMENT ******** START OF CODE **************;                                            04677500 T  0006
                IF XTA ← LABL = BLANKS THEN FLAG(18) ELSE                                                  04678000 T  0009
                IF D← SEARCH(LABL) = 0 THEN                                                                04678500 T  0011
                   D← ENTER(0 & FORMATID[TOCLASS], LABL) ELSE                                              04679000 T  0014
                   IF GET(D).CLASS ≠ FORMATID THEN BEGIN D←0;FLAG(143) END;                                 04679400 T  0019
                IF XREF THEN ENTERX(LABL,1&FORMATID[TOCLASS]);                                             04679500 T  0021
                LABL ← BLANKS;                                                                             04680000 T  0022
                NCR ← BACKNCR(NCR);                                                                        04680500 T  0024
                WSA[0] ← TOTAL ← 1;                                                                        04680700 T  0026
                ROUND1: XTA ← BLANKS;                                                                      04681000 T  0026
                ROUND: NCR ← GETCHAR(NCR,T);                                                               04681002 T  0029
                       IF T = "]" THEN GO TO NOPLACE;                                                      04681004 T  0030
                       XTA ← T & XTA[12:18:30];                                                            04681010 T  0032
                       T←IF (T="&" OR T="<") AND                                                           04681020 T  0033
                          (HOLTOG OR NOT(HF OR STRINGF)) THEN "+" ELSE                                     04681030 T  0037
                       IF (T="@" OR T=":") AND (HOLTOG OR NOT HF) THEN """                                 04681060 T  0041
                                ELSE T;                                                                    04681100 T  0042
                       IF NOT (HF OR STRINGF) THEN                                                         04681120 T  0043
                          IF T = " " THEN GO TO ROUND                                                      04681140 T  0045
                          ELSE IF T="," THEN GO SL ;                                                       04683000 T  0046
                       IF HF THEN                                                                          04683500 T  0047
                       BEGIN                                                                               04684000 T  0047
                           STORECHAR(WSA[TOTAL],I,T);                                                      04684500 T  0049
                           IF HF ← REPEAT ← REPEAT - 1 ≠ 0 THEN                                            04685000 T  0051
                           IF I ← I+1 = 8 THEN                                                             04685500 T  0053
                           BEGIN IF TOTAL ← TOTAL +1 > LSTMAX THEN GO TO NUL;                              04686000 T  0056
                                 WSA[TOTAL] ← I ← 0; GO TO ROUND;                                          04686500 T  0058
                           END ELSE GO TO ROUND ELSE GO TO NUL1;                                           04687500 T  0058
                       END;                                                                                04687510 T  0058
                      IF NOT STRINGF THEN
```

```
IF SLCNT > 0 THEN                                                          04687520 T  0059
IF T = "/" THEN BEGIN SLCNT ← SLCNT+1; GO TO ROUND; END                    04687530 T  0060
ELSE                                                                       04687550 T  0063
BEGIN WSA[TOTAL] ← 0 & SLCNT[TOREPEAT] & SLASH[TOCODE];                    04687560 T  0063
      IF NOT STR THEN                                                      04687580 T  0067
      IF REPEAT < 16 AND WSA[TOTAL-1].[42:6] = 0 THEN                      04687590 T  0068
        WSA[TOTAL←TOTAL-1] ← WSA[TOTAL] & SLCNT[42:44:4]                   04687600 T  0071
                  & 1[46:47:1];                                            04687620 T  0075
       COMMAS←DOLLARS←BOOLEAN(SLCNT←0);  NCR←BACKNCR(NCR) ;                04687650 T  0076
      GO TO NUL1;                                                          04687655 T  0080
END;                                                                       04687660 T  0080
 IF NOT QF THEN IF T = """ THEN IF STRINGF ← NOT STRINGF THEN              04688000 T  0080
  BEGIN IF CODE > 4 THEN BEGIN STRINGF ← FALSE;                            04688500 T  0083
     NCR ← BACKNCR(NCR); GO TO ENDER END;                                  04689000 T  0086
    SAVTOTAL ← TOTAL; J←0; I←3; QF ← TRUE;                                 04689500 T  0088
    WSA[TOTAL] ← 0 & HPHASE[TOCODE];                                       04690000 T  0091
    GO TO ROUND;                                                           04690500 T  0093
  END ELSE                                                                 04691000 T  0094
   BEGIN                                                                   04691500 T  0094
     WSA[SAVTOTAL] ← WSA[SAVTOTAL] & J[TOREPEAT];                          04692000 T  0094
      IF  I = 0 THEN TOTAL ← TOTAL - 1;                                    04693000 T  0097
     CODE ← HPHASE;                                                        04693200 T  0099
     GO TO ENDER;                                                          04693500 T  0100
   END;                                                                    04694000 T  0101
 IF STRINGF THEN                                                           04694500 T  0101
  BEGIN                                                                    04695000 T  0101
    STORECHAR(WSA[TOTAL],I,T);                                             04695500 T  0101
    J ← J + 1; QF ← FALSE;                                                 04696000 T  0103
    IF I ← I+1 = 8 THEN                                                    04696500 T  0105
  BEGIN                                                                    04697000 T  0107
      IF TOTAL ← TOTAL +1 > LSTMAX THEN GO TO NUL;                         04697500 T  0107
      T ← WSA[TOTAL] ← 0;                                                  04698000 T  0109
    END;                                                                   04698500 T  0111
      GO TO ROUND;                                                         04699000 T  0111
    END;                                                                   04699500 T  0112
CASE T OF                                                                  04707000 T  0112
BEGIN                                                                      04707500 T  0112
PRT(751) = *CASE STATEMENT DESCRIPTOR*
     BEGIN ZF ← TRUE;                                  % 0                 04708000 T  0112
      NUM: DECIMAL ← 10 × DECIMAL + T;                                     04708500 T  0113
           IF ASK THEN                                                     04708510 T  0115
              BEGIN FLAG(183) ;                                  %111-     04708515 P  0116
FL:           DO BEGIN NCR←GETCHAR(NCR,T); XTA←T&XTA[12:18:30] END         04708520 T  0117
              UNTIL T≠"*" AND T>9 AND T≠" " ;                              04708525 T  0121
              NCR←BACKNCR(NCR);  XTA←BLANKS&XTA[18:12:30] ;                04708530 T  0125
              END                                                          04708535 T  0128
           ELSE                                                            04708540 T  0128
           IF DECIMAL>4090 THEN BEGIN FLAG(172); DECIMAL←1 END ;           04708550 T  0128
           IF CODE = 0 THEN REPEAT ← DECIMAL                               04709500 T  0131
           ELSE IF PF THEN BEGIN IF DECIMAL>WIDTH AND WIDTH≠0 AND CODE≠    04710000 T  0133
                VPHRASE THEN FLAG(129) END ELSE WIDTH←DECIMAL ;            04710500 T  0138
           GO TO ROUND;                                                    04712000 T  0142
     END;                                                                  04712500 T  0142
     GO TO NUM; GO TO NUM; GO TO NUM;                  % 1 2 3             04713000 T  0143
     GO TO NUM; GO TO NUM; GO TO NUM;                  % 4 5 6             04713500 T  0144
     GO TO NUM; GO TO NUM; GO TO NUM;                  % 7 8 9             04714000 T  0146
     ; ; ; ; ; ;                                      % # @ Q : > ≥       04714500 T  0147
```

```
BEGIN PLUSP ← TRUE; GO TO ROUND; END;                    % +                    04714600 T  0147
BEGIN CODE ← APHASE; GO TO NOEND END;                    % A                    04715000 T  0149
;                                                        % B                    04715500 T  0151
BEGIN CODE ← CPHASE; GO TO NOEND END;                    % C                    04715600 T  0151
BEGIN CODE ← DPHASE; GO TO NOEND END;                    % D                    04716000 T  0152
BEGIN CODE ← EPHASE; GO TO NOEND END;                    % E                    04716500 T  0154
BEGIN CODE ← FPHASE; GO TO NOEND END;                    % F                    04717000 T  0156
BEGIN CODE ← GPHASE; GO TO NOEND END;                    % G                    04717500 T  0158
BEGIN IF REPEAT = 0 THEN FLOG(130);                      % H                    04718000 T  0159
      IF ASK THEN BEGIN FLOG(32 ); GO SEMIC END ;                               04718100 T  0161
      HF ← TRUE; I ← 3; CODE ← HPHASE;                                          04719000 T  0163
      WSA[TOTAL] ← 0 & HPHASE[TOCODE] & REPEAT[TOREPEAT];                       04719500 T  0166
      GO TO ROUND;                                                              04720000 T  0169
END;                                                                            04720500 T  0169
BEGIN CODE ← IPHASE; GO TO NOEND END;                    % I                    04721000 T  0170
BEGIN IF CODE < 11 OR CODE=15 THEN FLOG(134);            % .                    04721500 T  0172
      IF CODE=0 OR PF THEN FLOG(32) ;                                           04722000 T  0175
      PF←TRUE;  DECIMAL←0;  ASK←ZF←FALSE ;                                      04722500 T  0177
      GO TO ROUND;                                                              04723000 T  0180
END;                                                                            04723500 T  0180
GO TO RP;                                                % [                    04724000 T  0181
;                                                        % &                    04724500 T  0181
LP;                                                                             04725000 T  0181
BEGIN IF CODE ≠ 0 THEN FLOG(32);                         % (                    04725500 T  0182
      IF ASK THEN REPEAT←4095;  IF REPEAT=0 AND ZF THEN FLAG(173) ;             04725550 T  0184
      NAMLIST[SAVLASTLP ← PARENCT ← PARENCT+1] ← 0 & TOTAL[TOWIDTH]             04726000 T  0187
      &(IF REPEAT≤0 AND PARENCT>1 THEN 1 ELSE REPEAT)[TOREPEAT] ;               04726500 T  0190
      IF ASK THEN                                                               04726510 T  0195
         BEGIN ASK←VRB←FALSE ;                                                  04726520 T  0196
         WSA[TOTAL]←32&LPPHRASE[TOCODE]&4095[TOREPEAT] ;                        04726530 T  0197
         IF (TOTAL←TOTAL+1)>LSTMAX THEN GO NUL ;                                04726540 T  0201
         END ;                                                                  04726550 T  0203
      ZF←BOOLEAN(REPEAT←DECIMAL←0) ;                                            04727500 T  0203
      STR ← TRUE;                                                               04727600 T  0205
      GO TO ROUND1;                                                             04728000 T  0205
ENDI                                                                            04728500 T  0208
; ; ;                                                    % < ← ×              04729000 T  0208
BEGIN CODE←JPHASE; WIDTH←"1; GO NOEND END ;              % J                    04729500 T  0208
BEGIN                                                    % K                    04730000 T  0211
IF COMMAS OR CODE≠0 THEN BEGIN FLAG(32); COMMAS←TRUE END                        04730100 T  0211
ELSE BEGIN COMMAS←TRUE ;                                                        04730110 T  0214
KK:   DO BEGIN NCR←GETCHAR(NCR,T); XTA←T&XTA[12:18:30] END                       04730120 T  0215
      UNTIL T≠" " ;                                                             04730125 T  0219
      IF (T<17 OR (T>25 AND T<33) OR (T>42 AND T<50) OR T>57)                   04730130 T  0221
      THEN BEGIN FLAG(32) ;                                                     04730135 T  0226
           IF T="*" OR T<10 THEN BEGIN DECIMAL←1; GO FL END ;                   04730137 T  0228
           END ;                                                                04730139 T  0231
      NCR←BACKNCR(NCR);  XTA←BLANKS&XTA[18:12:30] ;                             04730140 T  0231
      END ;                                                                     04730145 T  0235
GO ROUND ;                                                                      04730150 T  0235
END OF K ;                                                                      04730160 T  0235
BEGIN CODE ← LPHASE; GO TO NOEND; END;                   % L                    04730500 T  0236
; ;                                                      % M N                  04731000 T  0237
BEGIN CODE ← OPHASE; GO TO NOEND; END;                   % O                    04731500 T  0237
BEGIN WSA[TOTAL] ← 0 & PPHASE[TOCODE]                    % P                    04732000 T  0239
       & REAL(VRB)[42:47:1]                                                     04732100 T  0241
       & REAL(MINUSP)[TOSIGN] & REPEAT[TOWIDTH]&1[TOREPEAT];                    04732500 T  0242
```

```
      MINUSP ← PLUSP ← FALSE;                                                    04733000 T  0245
      IF (DECIMAL = 0 AND NOT ZF) THEN FLOG(131);                                04733500 T  0247
      GO TO NUL1;                                                                04734500 T  0249
ENd;                                                                             04735000 T  0250
; ;                                                 % Q R                        04735500 T  0250
BEGIN IF DOLLARS OR CODE≠0 THEN FLAG(32)            % $                          04735600 T  0250
ELSE BEGIN DOLLARS←TRUE; GO KK END ;                                             04735610 T  0253
DOLLARS←TRUE;  GO ROUND ;                                                        04735620 T  0255
ENd OF DOLLAR SIGN ;                                                             04735630 T  0256
IF NOT ASK THEN                                     % *                          04735700 T  0256
    BEGIN                                                                        04735710 T  0257
     IF ZF OR DECIMAL NEQ 0 THEN FLAG(183); DECIMAL:=4095;     %111-             04735715 P  0257
    IF CODE=0 THEN REPEAT←DECIMAL                                                04735720 T  0261
    ELSE IF NOT PF THEN WIDTH←DECIMAL ;                                          04735730 T  0262
    VRB := ASK := LISTEL := TRUE; GO ROUNd;                    %101-             04735740 P  0266
     END ELSE BEGIN DECIMAL:=4095; FLAG(183); GO FL END ;      %111-             04735750 P  0269
BEGIN MINUSP ← TRUE; GO TO ROUND; END;              % -                          04736000 T  0273
RP:                                                                              04736500 T  0275
BEGIN IF FIELD THEN BEGIN NCR ← BACKNCR(NCR);       % )                          04737000 T  0276
        GO TO ENDER; END;                                                        04737500 T  0278
     IF DECIMAL ≠ 0 THEN FLAG(32);                                               04737600 T  0279
      I ← IF PARENCT = 1 THEN IF SAVLASTLP > 1 THEN 2 ELSE 1                     04738500 T  0281
           ELSE PARENCT;                                                         04739000 T  0284
      WSA[TOTAL]←(J←NAMLIST[I])&(TOTAL+1-J←J,[18:12])[TOLINK]                    04739500 T  0285
         & (IF PARENCT ← PARENCT-1 = 0 THEN 77 ELSE 0)[TODECIMAL];               04740000 T  0290
      IF WSA[J],[1:5]=LPPHRASE AND PARENCT≠0 THEN                                04740100 T  0294
         BEGIN WSA[J],[18:12]←TOTAL-J; WSA[TOTAL],[18:12]←TOTAL-J ;              04740110 T  0297
         END ;                                                                   04740115 T  0302
      NAMLIST[I],[6:12] ← 0;                                                     04740500 T  0303
      CODE ← HPHASE;                                                             04740600 T  0306
      GO TO NUL1;                                                                04742000 T  0307
     ENd;                                                                        04742500 T  0307
     ; ;                                            % ; LEQ                      04743000 T  0308
     GO TO ROUND;                                   % BLANKS                     04743500 T  0308
     BEGIN SLCNT ← 1;                               % /                          04744000 T  0308
SL:        IF CODE=0 THEN IF ASK OR ZF OR DECIMAL≠0 THEN                         04744100 T  0309
              BEGIN FLAG(32);  ASK←ZF←BOOLEAN(DECIMAL←0) END ;                   04744110 T  0313
           IF CODE<5 THEN IF T="," THEN GO ROUND1 ELSE GO ROUND ELSE GO          04744500 T  0316
                                                          ENDER ;                04744505 T  0319
     ENd;                                                                        04745000 T  0319
     ;                                              % S                          04745500 T  0319
     BEGIN IF REPEAT ≠ 0 THEN FLAG(32);             % T                          04746000 T  0319
           CODE ← TPHASE;                                                        04746050 T  0321
           GO TO NOEND;                                                          04746100 T  0322
     END;                                                                        04746150 T  0322
     ;                                              % U                          04746500 T  0323
     BEGIN VRB←TRUE; CODE←VPHRASE; WIDTH←-1; GO NOEND END ; % V                  04746750 T  0323
     ;                                              % W                          04746800 T  0326
     BEGIN IF REPEAT = 0 THEN FLOG(130);            % X                          04747000 T  0326
           IF STR THEN                                                           04747200 T  0328
NEWWD:        WSA[TOTAL] ← 0 & XPHASE[TOCODE] & REPEAT[TOWIDTH]                  04747400 T  0329
                             & 1[TOREPEAT]                                       04747600 T  0332
                              & REAL(VRB)[42:47:1]                               04747700 T  0333
           ELSE                                                                  04747800 T  0334
              BEGIN                                                              04748000 T  0335
                   IF (J←WSA[TOTAL-1]),[42:6]>0 OR (I←J,[1:5])=RTPARN            04748800 T  0335
                      OR (REPEAT≥32 AND I≠XPHASE) THEN GO NEWWD ;                04749000 T  0339
```

```
                    IF I=XPHASE AND (I←J,[18:12]+REPEAT)≤4090 THEN
                         WSA[TOTAL←TOTAL+1] ← J & I[TOWIDTH]
                       ELSE IF REPEAT ≥ 32 THEN GO TO NEWWD
                    ELSE WSA[TOTAL←TOTAL+1] ← J & REPEAT[TONUM]
                                               & 1[TOCNTRL];
               END;
               GO TO NUL1;
          END;
          ; ;                                                  % Y Z
          GO SL ;                                              % ,
          GO TO LP;                                            % %
          ; ; ;                                                % ≠ = ] "
END OF CASE STATEMENT;


FLOG(132);    %  ILLEGAL CHARACTER;
GO TO FALL;
ENDER: IF CODE > 4 THEN
       BEGIN IF WIDTH=0 THEN FLAG(130) ;
             IF CODE=VPHRASE THEN
                BEGIN
                IF WIDTH=-1 THEN IF PF THEN FLAG(130)ELSE WIDTH←
                DECIMAL←4094 ELSE
                IF NOT PF THEN DECIMAL←4094 ;
                END
             ELSE
             IF CODE > 10 AND CODE ≠ 15 THEN
               IF (DECIMAL = 0 AND NOT ZF) OR NOT PF THEN FLAG(133)
             ELSE ELSE DECIMAL ← 0;
             IF REPEAT=0 THEN REPEAT←1 ;
             IF WIDTH=-1 THEN WIDTH←0 ;
             WSA[TOTAL] ← 0 & CODE[TOCODE] & WIDTH[TOWIDTH]
                            & REPEAT[TOREPEAT] & DECIMAL[TODECIMAL]
                            & REAL(COMMAS) [44:47:1]
                            & REAL(VRB)[42:47:1]
                            & REAL(DOLLARS) [45:47:1];
       END ELSE IF DECIMAL ≠ 0 THEN FLAG(32);
NUL1:  IF PLUSP THEN FLAG(164);
       IF CODE≠VPHRASE THEN
       BEGIN
       IF DOLLARS AND(CODE < 9 OR CODE > 14) THEN FLAG(166);
       IF COMMAS AND NOT(CODE = 10 OR CODE = 12 OR CODE = 9)
          THEN FLAG(165);
       END ;
       VRB←
       ERRORTOG ← FIELD ← PF ← PLUSP ← DOLLARS ← COMMAS ← STR ← FALSE;
       IF CODE = HPHASE THEN STR ← TRUE;
       CODE ← REPEAT ← WIDTH ← 0;
       XTA ← BLANKS;
       GO TO FALL;
NOEND: IF FIELD THEN FLAG(32);
       IF CODE ≠ TPHASE THEN LISTEL ← TRUE ELSE REPEAT ← 1;
       IF REPEAT=0 AND ZF THEN FLAG(173) ;
       FIELD ← TRUE;
FALL:  IF MINUSP THEN BEGIN FLAG(32); MINUSP ← FALSE END;
       ASK←ZF←FALSE ;
NUL:   DECIMAL ← 0;
```

```
04749200 T  0343
04749400 T  0346
04749600 T  0349
04749800 T  0352
04750000 T  0356
04751000 T  0357
04752000 T  0357
04752500 T  0358
04753000 T  0358
04753500 T  0358
04754000 T  0359
04754500 T  0359
04755000 T  0359
START OF SEGMENT **********  137
 137 IS   64 LONG, NEXT SEG  134
04755500 T  0360
04756000 T  0360
04756500 T  0361
04757000 T  0362
04757400 T  0365
04757410 T  0366
04757420 T  0366
04757425 T  0370
04757430 T  0371
04757440 T  0374
04757450 T  0374
04757500 T  0374
04758000 T  0378
04758100 T  0382
04758400 T  0384
04758410 T  0386
04758500 T  0388
04759000 T  0391
04759100 T  0393
04759150 T  0394
04759200 T  0395
04760000 T  0397
04760500 T  0399
04760550 T  0401
04760560 T  0402
04760600 T  0402
04760700 T  0406
04760800 T  0408
04760850 T  0411
04760890 T  0411
04760900 T  0411
04760940 T  0415
04760980 T  0417
04761000 T  0419
04761500 T  0419
04762000 T  0420
04762500 T  0422
04762510 T  0425
04763000 T  0428
04763500 T  0429
04764000 T  0431
04764500 T  0432
```

```
                    IF PARENCT = 0 THEN BEGIN SCN ← 1; GO TO SEMIC END;                    04765000 T  0433
                     IF CODE < 5 THEN                                                      04765500 T  0436
                    IF TOTAL ← TOTAL+1 > LSTMAX THEN                                       04766000 T  0437
                        BEGIN FLOG(78);TOTAL ← TOTAL-2; GO TO SEMIC; END;                  04766500 T  0439
                GO TO ROUND;                                                               04767000 T  0442
                NOPLACE; IF(DCINPUT OR FREEFTOG) AND (STRINGF OR HF) THEN FLOG(150);       04767500 T  0442
                        IF TSSEDITOG THEN IF (STRINGF OR HF) AND NOT DCINPUT               04768000 T  0447
                        THEN TSSED(XTA,1);                                                 04768500 T  0449
                        IF CONTINUE THEN IF READACARD THEN                                 04769000 T  0451
                           BEGIN IF LISTOG THEN PRINTCARD; GO TO ROUND; END;               04769500 T  0453
                SCN ← 0; NEXT ← SEMI;                                                      04770000 T  0456
                SEMIC;                                                                     04770500 T  0457
                IF SCN ≠ 1 THEN SCAN;                                                      04771000 T  0458
                IF STRINGF THEN FLAG (22);                                                 04771500 T  0459
                IF NOT LISTEL THEN WSA[0] ← 0;                                             04772000 T  0461
                IF PARENCT ≠ 0 THEN FLAG(IF PARENCT < 0 THEN 9 ELSE 8);                    04772500 T  0463
                IF D ≠ 0 THEN PRTSAVER(D,TOTAL+1,WSA);                                     04772600 T  0467
                IF DEBUGTOG THEN BEGIN                                                     04772605 T  0471
                     WRITE(LINE,FM) ;                                                      04772610 T  0471
                      FOR I←0 STEP 1 UNTIL TOTAL DO BEGIN                                  04772615 T  0474
                    WRITE(LINE,[13]//,I,(J←WSA[I]),[1:5],J,[6:12],J,[18:12],J,[30:12],     04772620 T  0476
PRT(752) = *LIST, LABEL, OR SEGMENT DESCRIPTOR*
                                    J,[41:1],J,[42:4],J,[42:5],J,[44:1],J,[45:1],          04772625 T  0492
                                    J,[46:1],J,[46:2],J,[47:1]) ;                          04772630 T  0505
                     IF J,[1:5]=2 THEN I←I+(J,[6:12]+2),[36:9] ;                           04772635 T  0515
                      END ;                                                                04772640 T  0519
                     WRITE(LINE[DBL]) ;                                                    04772645 T  0522
                     END OF DEBUGSTUFF ;                                                   04772650 T  0526
                END FORMATER;                                                              04773000 T  0526
```

134 IS 535 LONG, NEXT SEG 6

```
                PROCEDURE EXECUTABLE;                                                      04773010 T  0556
                BEGIN LABEL XIT; REAL T, J, TS, P;                                         04773020 T  0556
```

START OF SEGMENT ********** 138

```
STACK(F+2) = T
STACK(F+3) = J
STACK(F+4) = TS
STACK(F+5) = P
                     IF SPLINK < 0 THEN FLAG(12);                                          04773030 T  0000
                     IF LABL =  BLANKS THEN GO TO XIT;                                     04773040 T  0002
                     IF T ← SEARCH(XTA ← LABL) = 0 THEN                                    04773050 T  0003
                     T ← ENTER(-0 & LABELID[TOCLASS] & (ADR+1)[TOADDR] &                   04773060 T  0005
                        NSEG[TOSEGNO], LABL) ELSE                                          04773070 T  0009
                BEGIN  IF (P ← GET(T)).CLASS ≠ LABELID THEN                                04773080 T  0011
                      BEGIN FLAG(144); GO TO XIT END;                                      04773090 T  0014
                     IF P < 0 THEN BEGIN FLAG(20); GO TO XIT END;                          04773100 T  0015
                     TS ← P.ADDR;                                                          04773110 T  0018
                     WHILE TS ≠ 0 DO                                                       04773120 T  0019
                     BEGIN J ← GIT(TS); FIXB(TS+10000); TS ← J END;                        04773130 T  0021
                     PUT(T, P←-P & (ADR+1)[TOADDR] & NSEG[TOSEGNO]);                       04773140 T  0026
                     IF (T ← GET(T+2)).BASE ≠ 0 THEN                                       04773150 T  0030
                     T ← PRGDESCBLDR(2, T.BASE, (ADR+1).[36:10], NSEG);                    04773160 T  0033
                END;                                                                       04773170 T  0037
                     IF XREF THEN ENTERX(LABL,1&LABELID[TOCLASS]);                         04773175 T  0037
                XIT;                                                                       04773180 T  0040
                END EXECUTABLE;                                                            04773190 T  0040
```

138 IS 43 LONG, NEXT SEG 6

```
                PROCEDURE TOCOMMAND(N);   VALUE N;   REAL N;                               04774000 T  0556
```

```
PRT(753) = ↑OCOMMAND                                                                                    04775000 T  0556
                COMMENT  N     COMMAND                                                                  04776000 T  0556
                         0     READ                                                                     04777000 T  0556
                         1     WRITE                                                                    04778000 T  0556
                         2     PRINT                                                                    04779000 T  0556
                         3     PUNCH                                                                    04780000 T  0556
                         4     BACKSPACE                                                                04781000 T  0556
                                                                                                        04782000 T  0556
                         7     DATA;                                                                    04783000 T  0556
                BEGIN LABEL XIT,SUCH,LISTER,NOFORM,FORMER,WRAP,DAAT,NF;                                 04784000 T  0556
                                                                                        START OF SEGMENT **********  139
                LABEL L↑STER1;                                                                          04784200 T  0000
                    BOOLEAN SUCHTOG, RDTRIN, FREEREAD;                                                  04785000 T  0000
STACK(F+2) = SUCHTOG
STACK(F+3) = RDTRIN
STACK(F+4) = FREFREAD
                      BOOLEAN FORMARY, NOFORMT;                                                         04785020 T  0000
STACK(F+5) = FORMARY
STACK(F+6) = NOFORMT
                    BOOLEAN NAMETOG;                                                                    04785050 T  0000
STACK(F+7) = NAMETOG
                DEFINE DATATOG = DATASTMTFLAG#;                                                         04785100 T  0000
                REAL T, ACCIDENT, EDITCODE;                                                             04786000 T  0000
STACK(F+10) = T
STACK(F+11) = ACCIDENT
STACK(F+12) = EDITCODE
                REAL DATAB;                                                                             04786100 T  0000
STACK(F+13) = DATAB
                PROCEDURE ACTIONLABELS(UNSEEN); VALUE UNSEEN; BOOLEAN UNSEEN;                           04787000 T  0000
PRT(754) = ACTIONLABELS
                BEGIN LABEL EOF,ERR,RATA,XIT,ACTION,MULTI;                                              04788000 T  0000
                                                                                        START OF SEGMENT **********  140
                    BOOLEAN BACK,GOTERR,GOTEOF;                                                         04789000 T  0000
STACK(F+2) = BACK
STACK(F+3) = GOTERR
STACK(F+4) = GOTEOF
                IF UNSEEN THEN SCAN;                                                                    04790000 T  0000
                EOF: IF GOTEOF THEN GO TO MULTI;                                                        04791000 T  0001
                     IF BACK ← NAME = "END    " THEN GO TO ACTION;                                      04792000 T  0003
                ERR: IF GOTERR THEN GO TO MULTI;                                                        04793000 T  0005
                     IF NAME ≠ "ERR    " THEN IF GOTEOF THEN                                            04794000 T  0006
                     BEGIN MULTI: XTA ← NAME;  FLOG(137);                                               04795000 T  0007
                                  GO TO XIT;                                                            04796000 T  0009
                     END ELSE GO TO RATA;                                                               04797000 T  0012
                ACTION: SCAN;                                                                           04798000 T  0012
                     IF NEXT = EQUAL THEN SCAN ELSE GO TO RATA;                                         04799000 T  0012
                     IF NEXT ≠ NUM THEN GO TO RATA;                                                     04800000 T  0014
                     IF XREF THEN ENTERX(NAME,0&LABELID[TOCLASS]);                                      04800100 T  0015
                     IF BACK THEN NX1 ← NAME ELSE NX2 ← NAME;                                           04801000 T  0018
                     SCAN; IF NEXT = RPAREN THEN GO TO XIT;                                             04802000 T  0021
                           IF NEXT = COMMA THEN SCAN ELSE GO TO RATA;                                   04803000 T  0022
                     IF BACK THEN                                                                       04804000 T  0024
                     BEGIN BACK ← NOT ( GOTEOF ← TRUE);                                                 04805000 T  0024
                           GO TO ERR;                                                                   04806000 T  0026
                     END;                                                                               04807000 T  0027
                     GOTERR ← TRUE;                                                                     04808000 T  0027
```

```
      GO TO EOF;
RATA: XTA ← NAME;  FLOG(0);
XIT:
END ACTIONLABELS;

IF DEBUGTOG THEN FLAGROUTINE(" !OCOM","MAND  ",TRUE ) ;
EODS←N≠7 ;
C2 ← IF N = 0 OR N = 7 THEN 1 ELSE 0;
SCAN; IF NEXT = SEMI THEN BEGIN FLOG(0); GO TO XIT END;
IF N = 7 THEN
BEGIN DATATOG ← TRUE;
      IF LOGIFTOG THEN FLAG(101);
         LABL ← BLANKS;
      IF SPLINK ≥ 0 THEN      %NOT BLOCK DATA STMT
      BEGIN
         IF DATAPRT=0 THEN BEGIN
              DATAPRT←PRTS←PRTS+1; ADJUST;
              DATASTRT←(ADR+1)&NSEG[TOSEGNO] END
         ELSE FIXB(DATALINK);
         EMITOPDCLIT(DATAPRT); EMITO(LNG);
         EMITB(-1, TRUE); DATAB ← LAX;
      END;
      GO TO DAAT;
END;
         EXECUTABLE;
EMITO(MKS);
IF N = 4 THEN
BEGIN
      INLINEFILE ;
      BEGIN EMITL(0);  EMITL(0);  EMITL(0);  EMITL(0);
            EMITL(5);  EMITL(0);  EMITL(0);
            EMITV(NEED(",FBINB",INTRFUNID));
      END;
      GO TO XIT;
END;
FDITCODE ← NX1 ← NX2 ← 0;
IF RDTRIN ←
        N = 0 THEN IF NEXT = LPAREN THEN GO TO SUCH
   ELSE EMITDESCLIT(FILECHECK(",5    ",2+17×REAL                    %503-
                                                        (REMOTETOG)))
ELSE IF N = 1 THEN IF NEXT ≠ LPAREN THEN FLAG(33)
                   ELSE GO TO SUCH
     ELSE IF N = 2 THEN                                             %503-
        EMITDESCLIT(FILECHECK(",6    ",2+17×REAL                    %503-
                                                        (REMOTETOG)))

                   ELSE EMITDESCLIT(FILECHECK(",PUNCH",0));
IF RDTRIN THEN EMITL(0) ELSE EMITPAIR(1,SSN);
GO TO FORMER;
SUCH:    SCAN;  RANDOMTOG←SUCHTOG←TRUE;  INLINEFILE ;
         RANDOMTOG←FREEREAD←FALSE ;
      IF NEXT = EQUAL THEN    %  RANDOM KEY
      BEGIN SCAN;
            IF EXPR(TRUE) > REALTYPE THEN FLAG(102);
            IF RDTRIN THEN EMITPAIR(1,ADD);
      END ELSE IF RDTRIN THEN EMITL(0) ELSE EMITPAIR(1,SSN);
      IF NEXT = RPAREN THEN GO TO NF;
```

```
                                                    04809000 T  0028
                                                    04810000 T  0028
                                                    04811000 T  0030
                                                    04812000 T  0031
                                          140 IS    34 LONG, NEXT SEG  139
                                                    04812010 T  0000
                                                    04812020 T  0002
                                                    04812050 T  0003
                                                    04813000 T  0007
                                                    04814000 T  0012
                                                    04815000 T  0012
                                                    04816000 T  0015
                                                    04816100 T  0017
                                                    04816110 T  0017
                                                    04816120 T  0018
                                                    04816130 T  0019
                                                    04816135 T  0020
                                                    04816136 T  0022
                                                    04816137 T  0024
                                                    04816140 T  0026
                                                    04816150 T  0027
                                                    04816160 T  0029
                                                    04817000 T  0029
                                                    04818000 T  0030
                                                    04819000 T  0030
                                                    04820000 T  0030
                                                    04825100 T  0031
                                                    04825200 T  0032
                                                    04826000 T  0032
                                                    04832000 T  0033
                                                    04833000 T  0036
                                                    04834000 T  0038
                                                    04835000 T  0039
                                                    04836000 T  0039
                                                    04837000 T  0042
                                                    04838000 T  0042
                                                    04839000 T  0043
                                                    04840000 T  0043
                                                    04841000 P  0046
                                                    04841100 T  0048
                                                    04842000 T  0049
                                                    04843000 T  0054
                                                    04844000 P  0055
                                                    04845000 P  0056
                                                    04845100 T  0058
                                                    04846000 T  0059
                                                    04847000 T  0059
                                                    04848000 T  0063
                                                    04849000 T  0068
                                                    04850000 T  0068
                                                    04850100 T  0072
                                                    04852000 T  0074
                                                    04853000 T  0075
                                                    04854000 T  0076
                                                    04855000 T  0078
                                                    04856000 T  0080
                                                    04857000 T  0084
```

```
    IF NEXT ≠ COMMA THEN BEGIN FLOG(114); GO TO XIT END;                                04858000 T  0085
    SCAN;                                                                               04859000 T  0087
    IF NEXT = ID THEN                                                                   04860000 T  0088
          IF NAME = "ERR    " OR NAME = "END    " THEN                                  04861000 T  0089
          BEGIN ACTIONLABELS(FALSE);                                                    04862000 T  0091
            NF: IF RDTRIN THEN EMITL(0) ELSE EMITPAIR(1,SSN);                           04863000 T  0092
                EMITL(0);                                                               04863100 T  0098
                NOFORMT ← TRUE;                                                         04863500 T  0098
                SCAN;  GO TO NOFORM;                                                    04864000 T  0099
          END;                                                                          04865000 T  0100
FORMER: IF ADR ≥ 4085 THEN                                                              04866000 T  0100
    BEGIN ADR ← ADR+1; SEGOVF END;                                                      04866100 T  0101
    IF NEXT = NUM THEN      % FORMAT NUMBER                                             04866200 T  0104
    BEGIN EDITCODE ← 1;                                                                 04867000 T  0104
          IF TEST ← LBLSHFT(NAME) ≤ 0 THEN                                              04868000 T  0106
          BEGIN FLAG(135); GO TO LISTER END;                                            04869000 T  0108
          IF I ← SEARCH(TEST) = 0 THEN  %  NEVER SEEN                                   04870000 T  0112
          OFLOWHANGERS(I←ENTER(0&FORMATID[TOCLASS], TEST)) ELSE                         04871000 T  0113
          IF GET(I).CLASS ≠ FORMATID THEN                                               04871100 T  0117
          BEGIN FLAG(143); GO TO LISTER END;                                            04871200 T  0119
          IF XREF THEN ENTERX(TEST,0&FORMATID[TOCLASS]);                                04873000 T  0121
          IF GET(I).ADDR = 0 THEN                                                       04874000 T  0124
          BEGIN EMITLINK((INFC ← GET(I + 2)).BASE);                                     04875000 T  0125
                PUT(I + 2,INFC&ADR[TOBASE]);                                            04876000 T  0129
                EMITL(0); EMITL(0); EMITO(NOP);                                         04877000 T  0131
          END ELSE                                                                      04878000 T  0133
          BEGIN EMITL(GET(I+ 2).BASE);                                                  04879000 T  0133
                EMITPAIR(GET(I).ADDR,LOD);                                              04880000 T  0136
          END;                                                                          04881000 T  0138
          GO TO LISTER;                                                                 04882000 T  0138
END ELSE IF RDTRIN THEN IF(FREEREAD := NEXT=SLASH) THEN GO TO LISTER                    04883000 T  0139
ELSE BEGIN IF NEXT NEQ ID THEN BEGIN FLOG(116);GO TO XIT; END;END                       04883100 T  0141
 ELSE IF NEXT NEQ ID THEN                                                               04883150 T  0144
  BEGIN IF NEXT = STAR THEN                                                             04883200 T  0146
        BEGIN NAMEDESC := TRUE; GLOBALNAME := TRUE;                                     04883300 T  0147
              TV := ENTER(0&LISTSID[TOCLASS],LISTID:=LISTID+1);                         04883350 T  0151
              SCAN;                                                                     04883400 T  0154
        END;                                                                            04883450 T  0155
        IF NEXT = LPAREN THEN                                                           04883500 T  0155
        BEGIN SCAN; IF EXPR(TRUE) GTR REALTYPE THEN FLAG(120) ;                         04883550 T  0156
              SCAN; END ELSE EMITL(0);                                                  04883600 T  0159
        IF GLOBALNAME AND (FREEREAD := NEXT = SLASH) OR FREEREAD THEN                   04883650 T  0161
            GO TO LISTER ELSE BEGIN FLOG(110); GO TO XIT; END;                          04883700 T  0164
  END;                                                                                  04883750 T  0166
      GETALL(I ← FNEXT,INFA,INFB,INFC);                                                 04884000 T  0166
      IF T ← INFA.CLASS = ARRAYID THEN  %  FORMAT ARRAY                                 04885000 T  0168
      BEGIN EDITCODE ← 1;                                                               04886000 T  0169
         FORMARY ← TRUE;                                                                04886050 T  0171
         T ← EXPR(FALSE);                                                               04886100 T  0171
         ADR ← ADR-1; % ELIMINATE XCH EMITTED BY EXPR                                   04887000 T  0173
         IF EXPRESULT ≠ ARRAYID THEN FLOG(116);                                         04887100 T  0174
         GO TO LISTER1; % SCAN ALREADY DONE IN EXPR                                     04888000 T  0176
      END ELSE                                                                          04889000 T  0176
      IF T = NAMELIST THEN                                                              04890000 T  0176
       BEGIN  NAMETOG := TRUE;                                                          04890100 T  0178
        IF INFA.ADDR = 0 THEN   %  REFERENCED, NOT DEF                                  04891000 T  0179
        BEGIN EMITLINK(INFC,BASE);                                                      04892000 T  0180
```

```
                PUT(I+ 2,(INFC ← INFC&ADR[TOBASE]));                                04893000 T  0182
                EMITL(0); EMITL(0); EMITO(NOP);                                     04894000 T  0184
          END ELSE                                                                  04895000 T  0187
          BEGIN EMITL(INFC,BASE);                                                   04896000 T  0187
                EMITPAIR(INFA,ADDR,LOD);                                            04897000 T  0188
          END                                                                       04898000 T  0190
        END                                                                         04898100 T  0190
      ELSE IF T = UNKNOWN THEN   %  ASSUME NAMELIST                                 04899000 T  0190
      BEGIN PUT(I,(INFA ← INFA&NAMELIST[TOCLASS]));                                 04900000 T  0191
            NAMETOG := TRUE;                                                        04900100 T  0194
            OFLOWHANGERS(I);                                                        04901000 T  0195
            EMITLINK(0);  PUT(I + 2,INFC&ADR[TOBASE]);                              04902000 T  0196
            EMITL(0); EMITL(0); EMITO(NOP);                                         04903000 T  0198
      END ELSE BEGIN XTA ← INFB; FLOG(116); GO TO XIT END;                           04904000 T  0201
      SCAN;                                                                         04905000 T  0203
      IF NEXT = COMMA THEN ACTIONLABELS(TRUE);                                      04906000 T  0204
      IF SUCHTOG THEN                                                               04907000 T  0206
         IF NEXT ≠ RPAREN THEN FLOG(108) ELSE SCAN;                                 04908000 T  0206
      IF NEXT ≠ SEMI THEN BEGIN FLOG(118); GO TO XIT END;                           04909000 T  0209
      EMITL(0);   EDITCODE ← 4;   EMITOPDCLIT(7);   EMITO(FTC);                     04910000 T  0212
      GO TO WRAP;                                                                   04911000 T  0215
LISTER:  SCAN;                                                                      04912000 T  0215
         IF FREEREAD THEN IF NOT RDTRIN THEN                                        04912100 T  0216
         BEGIN IF NEXT ≠ SLASH THEN EMITO(SSN) ELSE SCAN;                           04912200 T  0217
               IF NEXT = LPAREN THEN                                                04912300 T  0221
         BEGIN SCAN; IF EXPR(TRUE) > REALTYPE THEN FLAG(120);SCAN                   04912400 T  0222
               END ELSE EMITL(0);                                                   04912500 T  0225
         END;                                                                       04912600 T  0227
LISTER1:                                                                            04912700 T  0227
      IF SUCHTOG THEN                                                               04913000 T  0228
      BEGIN IF NEXT = COMMA THEN ACTIONLABELS(TRUE);                                04914000 T  0228
      IF NEXT = RPAREN THEN SCAN ELSE BEGIN FLOG(108); GO TO XIT END;               04915000 T  0230
      END ELSE IF NEXT=COMMA THEN SCAN ELSE IF RDTRIN THEN                          04916000 T  0234
         IF NEXT≠SEMI THEN FLOG(114);                                               04916100 T  0237
NOFORM: IF NEXT=SEMI THEN                                                           04917000 T  0239
      BEGIN IF FREEREAD THEN FLOG(061) ELSE EMITL(0); GO TO WRAP END;               04917100 T  0240
      IF (NEXT NEQ LPAREN) AND (NEXT NEQ ID) AND (NEXT NEQ STAR) THEN               04918000 T  0244
               GO TO XIT;                                                           04918100 T  0247
      EDITCODE ← EDITCODE + 2;                                                      04919000 T  0248
DAAT: EMITB(-1,FALSE);  LADR1 ← LAX;  ADJUST;  DESCREQ ← TRUE;                      04920000 T  0249
    IF ADR ≥ 4085 THEN                                                              04920100 T  0253
      BEGIN ADR ← ADR+1; SEGOVF; ADJUST END;                                        04920200 T  0254
      ACCIDENT ← PRGDESCBLDR(0,0,ADR,[36:10] + 1,NSEG);                             04921000 T  0256
      EMITOPDCLIT(19);   EMITO(GFW);                                                04922000 T  0259
      LITSTART ← ADR&NSEG[TOSEGNO];    ADJUST;                                      04923000 T  0261
      LA ← 0;   IOLIST(LA);                                                         04924000 T  0263
      EMITL(1);   EMITO(CHS);   EMITL(19);   EMITO(STD);                            04925000 T  0265
      EMITDESCLIT(19);   EMITO(RTS);                                                04926000 T  0268
      FIXB(LADR1);   DESCREQ ← FALSE;                                               04927000 T  0269
      IF DATATOG THEN                                                               04928000 T  0271
      BEGIN DATASET;                                                                04929000 T  0271
            IF NEXT = SLASH THEN SCAN ELSE                                          04930000 T  0272
            BEGIN FLOG(110); GO TO XIT END;                                         04931000 T  0274
      IF LSTA=0 THEN BEGIN BUMPPRT; LSTA←PRTS END;                                  04932000 T  0277
            IF (LSTMAX - LSTI) ≤ LSTS THEN                                          04933000 T  0283
            BEGIN WRITEDATA(LSTI,NXAVIL ← NXAVIL + 1,LSTP);                         04934000 T  0284
                  LSTA ← PRGDFSCBLDR(1,LSTA,0,NXAVIL);                              04935000 T  0287
```

```
            LSTI ← 0; BUMPPRT; LSTA←PRTS;                                                    04936000 T  0289
               END;                                                                          04937000 T  0294
               MOVEW(LSTT,LSTP[LSTI],(LSTS ← LSTS + 1),[36:6],LSTS);                         04938000 T  0294
               EMITO(MKS);   EMITL(LSTI);   EMITPAIR(LSTA,LOD);                              04939000 T  0298
               LSTI ← LSTI + LSTS;                                                           04940000 T  0300
               EMITPAIR(ACCIDENT,LOD); EMITOPDCLIT(7); EMITO(FTF);                           04941000 T  0302
               EMITL(6); EMITL(0); EMITL(0);                                                 04942000 T  0304
               EMITV(NEED(",FBINB",INTRFUNID));                                              04943000 T  0306
               IF NEXT = COMMA THEN                                                          04944000 T  0308
               BEGIN SCAN; GO TO DAAT END;                                                   04945000 T  0309
               IF SPLINK ≥ 0 THEN BEGIN                                                      04946000 T  0312
               EMITB(-1,FALSE); DATALINK←LAX;                                                04946500 T  0313
                    FIXB(DATAB) END;                                                         04946600 T  0315
               GO TO XIT;                                                                    04947000 T  0316
         END;                                                                                04948000 T  0316
         EMITPAIR(ACCIDENT,LOD); EMITOPDCLIT(7); EMITO(FTF);                                 04949000 T  0316
WRAP: IF NOT FREEREAD AND NOT NAMETOG THEN EMITL(EDITCODE);                                  04950000 T  0319
IF RDTRIN THEN                                                                               04951000 T  0321
BEGIN IF NX1 = 0 THEN EMITL(0) ELSE EMITLABELDESC(NX1);                                      04952000 T  0321
      IF NX2 = 0 THEN EMITL(0) ELSE EMITLABELDESC(NX2);                                      04953000 T  0325
      IF FREEREAD THEN EMITV(NEED(",FREFR", INTRFUNID))                                      04954000 T  0328
        ELSE IF NAMETOG THEN EMITV(NEED(",FINAM",INTRFUNID))                                 04954050 T  0330
        ELSE IF FORMARY THEN EMITV(NEED(",FTINT",INTRFUNID))                                 04954100 T  0334
        ELSE IF NOFORMT THEN EMITV(NEED(",FBINB",INTRFUNID))                                 04954150 T  0338
             ELSE EMITV(NEED(",FTNIN",INTRFUNID));                                           04954200 T  0342
END ELSE                                                                                     04955000 T  0346
IF FREEREAD THEN                                                                             04956000 T  0346
    BEGIN                                                                                    04956005 T  0348
    IF NAMEDESC THEN                                                                         04956010 T  0348
         BEGIN                                                                               04956020 T  0349
                    PRTSAVER(TV,NAMEIND+1,NAMLIST);                                          04956040 T  0350
                    EMITL(GET(TV+2),BASE);                                                   04956100 T  0352
                    EMITPAIR(GET(TV),ADDR,LOD);                                              04956200 T  0354
                    IF NAMLIST[0] = 0 THEN EMITL(0)                                          04956500 T  0356
                    ELSE EMITPAIR(GET(GLOBALSEARCH(",SUBAR")),ADDR,LOD);                     04956550 T  0358
                    NAMLIST[0] := NAMEIND := 0;                                              04956600 T  0362
               END ELSE BEGIN EMITL(0);EMITL(0);EMITL(0);END;                                04956650 T  0363
          EMITV(NEED(",FREWR",INTRFUNID))                                                    04956700 T  0368
          END ELSE IF NAMETOG THEN EMITV(NEED(",FONAM",INTRFUNID))                           04956750 T  0369
                    ELSE IF FORMARY THEN EMITV(NEED(",FTOUT",INTRFUNID))                     04956800 T  0373
                         ELSE BEGIN                                                          04956900 T  0377
                              IF NX1=0 THEN EMITL(0) ELSE EMITLABELDESC(NX1);                04956910 T  0380
                              IF NX2=0 THEN EMITL(0) ELSE EMITLABELDESC(NX2);                04956920 T  0383
                         IF NOFORMT THEN EMITV(NEED(",FBINB",INTRFUNID)) ELSE                04956925 T  0386
                              EMITV(NEED(",FTNOU",INTRFUNID));                               04956930 T  0388
                              END;                                                           04956940 T  0392
          XIT;                                                                               04957000 T  0392
               IF NAMEDESC THEN IF RDTRIN THEN FLAG(159)                                     04957050 T  0393
                           ELSE IF NOT FREEREAD THEN FLAG(160);                              04957055 T  0395
               DATATOG := FALSE;  NAMEDESC := FALSE; GLOBALNAME := FALSE;                    04957100 T  0399
          IF DEBUGTOG THEN FLAGROUTINE(" IOCOM","MAND  ",FALSE) ;                            04957110 T  0405
          END IOCOMMAND;                                                                     04958000 T  0407
                                                                                  139 IS  414 LONG, NEXT SEG    6
          PROCEDURE STMTFUN(LINK); VALUE LINK; REAL LINK;                                    04959000 T  0556
PRT(755) = STMTFUN                                                                           04960000 T  0556
          BEGIN                                                                              04961000 T  0556
               DEFINE PARAM = LSTT#;
```

```
                    REAL SAVEBRAD, I;                                                    04962000 T  0000
STACK(F+2) = SAVEBRAD
STACK(F+3) = I
                    REAL INFA, INFC, NPARMS, TYPE, PARMLINK, BEGINSUB, RETURN;           04963000 T  0000
STACK(F+4) = INFA
STACK(F+5) = INFC
STACK(F+6) = NPARMS
STACK(F+7) = TYPE
STACK(F+10) = PARMLINK
STACK(F+11) = BEGINSUB
STACK(F+12) = RETURN
                    LABEL XIT,TIX ;                                                      04964000 T  0000
                    IF SPLINK < 0 THEN FLAG(12);                                         04964100 T  0000
                    LABL ← BLANKS;                                                       04964200 T  0002
                    FILETOG ← TRUE; % PREVENTS SCANNER FROM ENTERING IDS IN INFO          04965000 T  0002
                    IF XREF THEN ENTERX(GET(LINK+1),0&STMTFUNID[TOCLASS]                 04965100 T  0003
                        &(GET(LINK))[21:21:3]);                                          04965200 T  0007
                    DO                                                                   04966000 T  0008
                BEGIN                                                                    04967000 T  0009
                    SCAN;                                                                04968000 T  0009
                    IF NEXT ≠ ID THEN BEGIN FLOG(107); GO TO XIT END;                    04969000 T  0009
                    PARAM[NPARMS←NPARMS+1] ← NAME;                                       04970000 T  0012
                    SCAN;                                                                04971000 T  0014
                END  UNTIL NEXT ≠ COMMA;                                                 04972000 T  0014
                    IF NEXT ≠ RPAREN THEN FLOG(108) ELSE SCAN;                           04973000 T  0016
                                                                                         04974000 T  0019
                    IF NEXT ≠ EQUAL THEN BEGIN FLOG(104); GO TO XIT END;                 04975000 T  0019
                    EMITB(-1, FALSE); SAVEBRAD ← LAX; % BRANCH AROUND ST FUN             04976000 T  0021
                    ADJUST;                                                              04977000 T  0023
                    BEGINSUB ← ADR+1;                                                    04978000 T  0024
                    BUMPLOCALS; EMITPAIR(RETURN←LOCALS+1536,STD);                        04979000 T  0025
                    FOR I ← NPARMS STEP -1 UNTIL 1 DO                                    04980000 T  0031
            BEGIN                                                                        04981000 T  0033
                    IF T ← SEARCH(PARAM[I]) ≠ 0 THEN                                     04982000 T  0033
                        TYPE ← GET(T).SUBCLASS ELSE                                      04983000 T  0035
                        IF T←PARAM[I].[12:6] < "I" OR T > "N" THEN                       04984000 T  0037
                        TYPE ← REALTYPE ELSE TYPE ← INTYPE;                              04985000 T  0040
                    EMITSTORE(          ENTER(0&VARID[TOCLASS]&1[TOTYPE]                 04986000 T  0043
                        &TYPE[TOSUBCL], PARAM[I]), TYPE);                                04987000 T  0046
                    IF XREF THEN ENTERX(NAME,0&VARID[TOCLASS]&TYPE[TOSUBCL]);            04987100 T  0048
            END;                                                                         04988000 T  0052
                    PARMLINK ← NEXTINFO-3;                                               04989000 T  0054
                    GETALL(LINK, INFA, XTA, INFC);                                       04990000 T  0055
                    FILETOG ← FALSE;                                                     04991000 T  0057
                    SCAN;                                                                04992000 T  0057
                    IF (TYPE←(INFA←GET(LINK)).SUBCLASS)=LOGTYPE OR TYPE=COMPTYPE OR      04993000 T  0058
                       (I←EXPR(TRUE))=LOGTYPE OR I=COMPTYPE THEN                         04993500 T  0061
                        BEGIN IF I≠TYPE THEN FLAG(139); GO TIX END ;                     04993510 T  0065
                    IF TYPE=REALTYPE OR TYPE=INTYPE THEN                                 04993530 T  0068
                        BEGIN                                                            04993540 T  0069
                        IF I=DOUBTYPE THEN BEGIN EMITO(XCH); EMITO(DEL) END ;            04993550 T  0070
                        IF TYPE=INTYPE THEN IF I≠INTYPE THEN EMITPAIR(1,IDV) ;           04993560 T  0073
                        GO TIX ;                                                         04993570 T  0076
                        END ;                                                            04993580 T  0077
                    IF I≠DOUBTYPE THEN EMITPAIR(0,XCH) ;                                 04993590 T  0077
TIX;                                                                                     04993595 T  0079
```

```
                        EMITOPDCL;IT(RETURN) ;                                                  04994000 T  0080
                        EMITO(GFW);                                                             04995000 T  0080
                        FIXB(SAVEBRAD);                                                         04996000 T  0081
                        IF INFA.CLASS ≠ UNKNOWN THEN FLAG(140);                                 04997000 T  0082
                        PUT(LINK, -INFA & 1[TOTYPF] & NSEG[TOSEGNO]                             04998000 T  0084
                        & STMTFUNID[TOCLASS] & BEGINSUB[TOADDR]);                               04999000 T  0087
                                                                                                04999500 T  0090
                        PUT(LINK+2, -(0 & NPARMS[TONEXTRA] & ADR[TOBASE]                        05000000 T  0090
                             & PARMLINK[36:36:12]));                                            05001000 T  0092
                        PARMLINK ← PARMLINK+4;                                                  05002000 T  0094
                        FOR I ← 1 STEP 1 UNTIL NPARMS DO                                        05003000 T  0095
                             PUT(PARMLINK ← PARMLINK-3, ".....");                               05004000 T  0097
                   XIT:                                                                         05005000 T  0101
                        FILETOG ← FALSE;                                                        05006000 T  0102
                   END STMTFUN;                                                                 05007000 T  0102
                                                                                          141 IS  108 LONG, NEXT SEG    6
                   PROCEDURE ASSIGNMENT;                                                        05008000 T  0556
PRT(756) = ASSIGNMENT
                   BEGIN                                                                        05009000 T  0556
                        LABEL XIT;                                                              05010000 T  0556
                                                                                         START OF SEGMENT **********  142
                   BOOLEAN CHCK;                                                                05010500 T  0000
STACK(F+2) = CHCK
                   BOOLEAN I;                                                                   05010600 T  0000
STACK(F+3) = I
                   IF DEBUGTOG THEN FLAGROUTINE(" ASSIG","NMENT ",TRUE ) ;                      05011000 T  0000
                        FX1 ← FNEXT;                                                            05012000 T  0002
                        SCAN;                                                                   05013000 T  0002
                        IF NEXT = LPAREN THEN                                                   05014000 T  0003
                   BEGIN                                                                        05015000 T  0004
                   CHCK←TRUE;                                                                   05015500 T  0004
                        IF GET(FX1).CLASS = UNKNOWN THEN                                        05016000 T  0005
                        IF EODS THEN                                                            05017000 T  0007
                           BEGIN  XTA ← GET(FX1+1);  FLOG(035) ;                                05017010 T  0007
                           PUT(FX1,GET(FX1) & ARRAYID[TOCLASS]) ;                               05017020 T  0010
                           PUT(FX1+2,GET(FX1+2) & 1[TONEXTRA]) ;                                05017030 T  0013
                           END                                                                  05017040 T  0016
                        ELSE BEGIN STMTFUN(FX1);  GO TO XIT END ;                               05017050 T  0016
                        IF XREF THEN ENTERX(GET(FX1+1),1&GET(FX1) [15:15:9]);                   05017055 T  0021
                        EODS ← TRUE ;                                                           05017060 T  0025
                        EXECUTABLE;                                                             05017100 T  0026
                        SCAN;                                                                   05018000 T  0026
                        I ← SUBSCRIPTS(FX1,2);                                                  05019000 T  0027
                        SCAN;                                                                   05020000 T  0028
                   END ELSE                                                                     05021000 T  0029
                   BEGIN                                                                        05022000 T  0029
                        EODS←TRUE ;                                                             05022010 T  0029
                        EXECUTABLE;                                                             05022100 T  0030
                        IF T ← GET(FX1).CLASS = ARRAYID THEN                                    05023000 T  0031
                        BEGIN XTA ← GET(FX1+1); FLAG(74) END;                                   05024000 T  0033
                        MOVEW(ACCUM[1],HOLDID[0],0,3);                                          05025000 T  0036
                        IF XREF THEN IF HOLDID[0].[12:12] ≠ "DO" THEN                           05025100 T  0038
                          ENTERX(GET(FX1+1),1&GET(FX1)[21:21:3]&VARID[TOCLASS]);                05025200 T  0040
                   END;                                                                         05026000 T  0045
                        IF NEXT ≠ EQUAL THEN BEGIN FLAG(104); GO TO XIT END;                    05027000 T  0045
                        SCAN;                                                                   05028000 T  0049
                        IF NEXT=SEMI OR NEXT=COMMA THEN BEGIN FLOG(0); GO TO XIT; END;          05028010 T  0049
```

```
          FX2 ← EXPR(TRUE);                                                                  05029000 T  0053
          IF NEXT NEQ COMMA THEN IF HOLDID[0] = "DO" THEN IF XREF THEN                       05029200 T  0054
            ENTERX(HOLDID[0] ,1&GET(FX1)[21:21:3]&VARID[TOCLASS]);                           05029400 T  0057
          IF NEXT = COMMA THEN IF CHCK THEN FLOG(56) ELSE                                    05030000 T  0061
          IF HOLDID[0],[12:12] ≠ "DO" THEN FLOG(56) ELSE                                     05030100 T  0064
     BEGIN                                                                                   05031000 T  0068
          IF LOGIFTOG THEN FLAG(101);                                                        05032000 T  0071
          IF FX2 > REALTYPE THEN FLAG(102);                                                  05033000 T  0073
          IF DT ← DT+1 > MAXDOS THEN BEGIN DT ← 1; FLAG(138) END;                            05034000 T  0075
          EMITN(FX1← CHECKDO);                                                               05035000 T  0078
          EMITO(STD);                                                                        05036000 T  0080
           SCAN;                                                                             05037000 T  0081
           IF NEXT=SEMI THEN BEGIN FLAG(36); GO TO XIT END;                                  05038000 T  0081
           IF (ACCUM[0] = ",     " OR ACCUM[0] = ";     ") AND                               05038900 T  0084
           GLOBALNEXT=NUM AND ABS(FNEXT) > 1023 THEN                                         05039000 T  0086
                BEGIN                                                                        05040000 T  0088
                IF EXPR(TRUE) > REALTYPE THEN FLAG(102);                                     05041000 T  0089
                IDINFO:=REALID;FNEXT:=ENTER(IDINFO,"2FNV00"&DT[36:36:12]);                   05041100 T  0091
                EMITN(FNEXT:=GETSPACE(FNEXT)); EMITO(STD);                                   05041200 T  0094
                EMITB(-1,FALSE); LADR1:=LAX; ADJUST;                                         05041300 T  0097
                LADR2 ← (ADR+1) & NSEG[TOSEGNO]; EMITV(FNEXT);                               05041400 T  0099
                END                                                                          05041450 T  0102
          ELSE BEGIN                                                                         05041500 T  0102
                EMITB(-1,FALSE); LADR1:=LAX; ADJUST;                                         05041600 T  0107
                LADR2:=(ADR+1)&NSEG[TOSEGNO];                                                05041700 T  0109
                IF EXPR(TRUE) > REALTYPE THEN FLAG(102) ;                                    05041800 T  0111
                END ;                                                                        05042000 T  0114
          EMITO(GRTR);                                                                       05043000 T  0114
          EMITB(-1, TRUE);                                                                   05044000 T  0115
          LADR3 ← LAX;                                                                       05045000 T  0116
          EMITB(-1, FALSE);                                                                  05046000 T  0117
          ADJUST;                                                                            05047000 T  0118
          DOTEST[DT] ← (ADR+1) & LAX[TOADDR] & NSEG[TOSEGNO];                                05048000 T  0118
          IF NEXT ≠ COMMA THEN EMITL(1) ELSE                                                 05049000 T  0122
     BEGIN                                                                                   05050000 T  0124
          SCAN;                                                                              05051000 T  0125
          IF NEXT=SEMI THEN BEGIN FLAG(36);  GO TO XIT END ;                                 05051100 T  0125
          IF EXPR(TRUE) > REALTYPE THEN FLAG(102);                                           05052000 T  0128
     END;                                                                                    05053000 T  0130
          EMITV(FX1);                                                                        05054000 T  0130
          EMITO(ADD);                                                                        05055000 T  0131
          EMITN(FX1);                                                                        05056000 T  0132
          EMITO(STN);                                                                        05057000 T  0132
          EMITB(LADR2, FALSE);                                                               05058000 T  0133
          FIXB(LADR1);                                                                       05059000 T  0134
          FIXB(LADR3);                                                                       05060000 T  0135
     END ELSE EMITSTORE(FX1, FX2);                                                           05061000 T  0136
     XIT;                                                                                    05062000 T  0137
IF DEBUGTOG THEN FLAGROUTINE(" ASSIG","NMENT ",FALSE) ;                                      05062010 T  0138
END ASSIGNMENT;                                                                              05063000 T  0140
                                                                                  142 IS  145 LONG, NEXT SEG    6
BOOLEAN PROCEDURE RINGCHECK;                                                                 05063100 T  0556
PRT(757) = RINGCHECK
COMMENT  THIS PROCEDURE PREVENTS THE POSSIBILITY OF DELINKING A                              05063110 T  0556
           HEADER FROM THE HEADER RING;                                                      05063120 T  0556
     BEGIN                                                                                   05063200 T  0556
          INTEGER I;                                                                         05063250 T  0556
```

```
STACK(F+3) = I
                        I←A;
                        DO
                        IF I ← GETC(I).ADDR = ROOT THEN RINGCHECK ← TRUE
                        UNTIL I = A;
                    END RINGCHECK;

                PROCEDURE SETLINK(INFADDR); VALUE INFADDR; INTEGER INFADDR;
PRT(760) = SETLINK
                COMMENT  THIS PROCEDURE LINKS AN ELEMENT TO ITS PREVIOUS HEADER;
                BEGIN
                    INTEGER LAST,I; REAL COML; LABEL XIT;

STACK(F+2) = LAST
STACK(F+3) = I
STACK(F+4) = COML
                XIT:
                    LAST ←(GETC(INFADDR).LASTC)=1;
                    FOR I ← INFADDR+2 STEP 1 UNTIL LAST
                    DO BEGIN IF GETC(I).CLASS = ENDCOM THEN I←GETC(I).LINK;
                             IF FX1 = (COML←GETC(I)).LINK THEN
                                   IF INFADDR←COML.LASTC=A THEN COM[PWI].LASTC←ROOT
                                   ELSE GO XIT ;
                       END;
                END SETLINK;

                PROCEDURE DIMENSION;
PRT(761) = DIMENSION
                BEGIN
                    LABEL L, LOOP, ERROR ;

                    BOOLEAN DOUBLED, SINGLETOG;
STACK(F+2) = DOUBLED
STACK(F+3) = SINGLETOG
                IF DEBUGTOG THEN FLAGROUTINE(" DIMEN","SION  ",TRUE ) ;
                      IF LOGIFTOG THEN FLAG(101);
                      LABL ← BLANKS;
                      IF NEXT=STAR THEN IF TYPE≠DOUBTYPE THEN
                          BEGIN
                          SCAN ;
                          IF NEXT=NUM AND NUMTYPE=INTYPE THEN
                              BEGIN
                              IF FNEXT=4 THEN
                                  BEGIN
                                  SINGLETOG ← TRUE;
                                  IF TYPE=COMPTYPE THEN FLAG(176);  GO L ;
                                  END ;
                              IF FNEXT=8 THEN
                                  BEGIN
                                  IF TYPE=REALTYPE THEN TYPE←DOUBTYPE
                                  ELSE IF TYPE≠COMPTYPE THEN FLAG(177) ;
                                  GO L ;
                                  END ;
                              END ;
                          FLAG(IF TYPE=REALTYPE THEN 178
                              ELSE 177-REAL(TYPE=COMPTYPE)) ;
```

```
START OF SEGMENT **********  143

        05063300 T  0000
        05063350 T  0000
        05063400 T  0001
        05063450 T  0004
        05063500 T  0005
143 IS    9 LONG, NEXT SEG    6
        05063600 T  0556

        05063601 T  0556
        05063610 T  0556
        05063620 T  0556
START OF SEGMENT **********  144



        05063625 T  0000
        05063630 T  0000
        05063640 T  0002
        05063650 T  0005
        05063660 T  0010
        05063670 T  0012
        05063680 T  0018
        05063710 T  0019
        05063730 T  0020
144 IS   23 LONG, NEXT SEG    6
        05064000 T  0556

        05065000 T  0556
        05066000 T  0556
                START OF SEGMENT **********  145
%109-           05066005 P  0000

        05066010 T  0000
        05067000 T  0002
        05067100 T  0004
        05067210 T  0004
        05067220 T  0006
        05067230 T  0007
        05067240 T  0007
        05067250 T  0009
        05067260 T  0010
        05067270 T  0010
%109-   05067275 C  0011
        05067280 T  0012
        05067290 T  0017
        05067300 T  0017
        05067310 T  0017
        05067320 T  0018
        05067330 T  0019
        05067340 T  0022
        05067350 T  0023
        05067360 T  0023
        05067370 T  0023
        05067380 T  0024
```

```
L:          NCR←REAL(NCR,[30:3]≠0)+3"677777"+NCR;  SCN←1;  SCAN ;          05067390 T  0027
            END ;                                                          05067420 T  0031
 LOOP:       DOUBLED←FALSE;                                                05068000 T  0031
     IF NEXT ≠ ID THEN  BEGIN FLOG(105); GO TO ERROR END;                  05068100 T  0031
     FX1 ← IF SINGLETOG THEN -FNEXT ELSE FNEXT;                  %109-     05069000 P  0036
     IF TYPE ≥ DOUBTYPE THEN   %  FIX ARRAY TYPE OFR                       05069100 T  0038
      PUT(FX1,GET(FX1)&TYPE[TOSUBCL]); % BOUNDS ROUTINE                    05069200 T  0039
     IF XREF THEN BEGIN INFA ← 0&GET(FX1)[15:15:9];                        05069300 T  0042
                        IF TYPE>0 THEN INFA.SUBCLASS←TYPE;                 05069400 T  0045
                  END;                                                     05069500 T  0048
     XTA ← INFB ← NAME;                                                    05070000 T  0048
     SCAN;                                                                 05071000 T  0049
     IF  XREF THEN                                                         05071100 T  0050
     BEGIN IF INFA.CLASS = UNKNOWN THEN                                    05071200 T  0050
              INFA.CLASS←IF NEXT=LPAREN THEN ARRAYID ELSE VARID;           05071300 T  0052
           ENTERX(INFB,INFA);                                              05071400 T  0056
     END;                                                                  05071500 T  0057
     IF NEXT=LPAREN THEN BEGIN SCAN; DOUBLED←BOUNDS(FX1) END ELSE          05072000 T  0057
     IF TYPE = -1 THEN FLOG(103);                                          05073000 T  0060
     GETALL(FX1, INFA, XTA, INFC);                                         05074000 T  0063
     IF TYPE > 0 THEN                                                      05075000 T  0064
     IF BOOLEAN(INFA.TYPEFIXED) THEN FLAG(31) ELSE                         05076000 T  0065
BEGIN                                                                      05077000 T  0067
     IF TYPE > LOGTYPE THEN                                                05078000 T  0068
     IF GET(FX1+2) <0 THEN                                                 05078200 T  0069
     BEGIN                                                                 05078400 T  0071
     IF NOT DOUBLED AND INFA.CLASS=1 THEN                                  05078500 T  0071
          BEGIN                                                            05078800 T  0073
          BUMPLOCALS;                                                      05079000 T  0074
          LENGTH←LOCALS + 1536;                                            05079200 T  0078
          PUT(FX1+2,INFC & LENGTH[TOSIZE]);                                05079400 T  0079
          END                                                              05079600 T  0081
     END ELSE IF NOT DOUBLED THEN                                          05079800 T  0081
          BEGIN IF INFC.SIZE > 16383 THEN FLAG(99);                        05079900 T  0083
          PUT(FX1+2,INFC & (2 × INFC.SIZE)[TOSIZE]);                       05080000 T  0086
          END;                                                             05080100 T  0089
     PUT (FX1,INFA & 1[TOTYPF] & TYPE[TOSUBCL]);                           05080500 T  0089
END;                                                                       05081000 T  0092
     IF INFA < 0 THEN FLAG(39) ELSE                                        05082000 T  0092
     IF TYPE = -2 THEN                                                     05083000 T  0094
BEGIN                                                                      05084000 T  0097
     BAPC(INFA&FX1[TOLINK]&1[TOCE]&ROOT[TOLASTC]);                         05085000 T  0097
     IF BOOLEAN(INFA.CE) THEN FLAG(2);                                     05086000 T  0101
     IF BOOLEAN(INFA.EQ) THEN                                              05086100 T  0103
BEGIN                                                                      05087000 T  0104
     COM[NEXTCOM.IR,NEXTCOM.IC].LASTC ← A ← INFA.ADDR;                     05088000 T  0104
     B←GETC(ROOT).ADDR ;                                                   05089000 T  0109
     SETLINK(A);                                                           05089050 T  0111
     IF NOT RINGCHECK THEN                                                 05089100 T  0112
BEGIN                                                                      05089200 T  0113
     COM[PWROOT].ADDR←GETC(A).ADDR ;                                       05090000 T  0113
     PUTC(A,GETC(A)&B[TOADDR]&7[TOSUBCL]) ;                                05091000 T  0118
END                                                                        05091100 T  0122
END  ELSE                                                                  05092000 T  0122
     PUT(FX1, INFA & 1[TOCE] & ROOT[TOADDR]);                              05093000 T  0122
     IF BOOLEAN(INFA.FORMAL) THEN FLAG(10);                                05094000 T  0125
END;                                                                       05095000 T  0127
```

```
                    IF ERRORTOG THEN                                                                05096000 T  0127
                ERROR;                                                                              05096100 T  0128
                    WHILE NEXT ≠ COMMA AND NEXT ≠ SEMI AND NEXT ≠ SLASH DO SCAN;                    05097000 T  0129
                    IF NEXT = COMMA THEN BEGIN SCAN; GO TO LOOP END;                                05098000 T  0133
            IF DEBUGTOG THEN FLAGROUTINE(" DIMEN","SION  ",FALSE) ;                                 05098010 T  0135
            END DIMENSION;                                                                          05099000 T  0137
                                                                                          145 IS  142 LONG, NEXT SEG    6
            PROCEDURE FORMALPP(PARMSREQ, CLASS); VALUE PARMSREQ, CLASS;                             05100000 T  0556
PRT(762) = FORMALPP
                    BOOLEAN PARMSREQ; REAL CLASS;                                                   05101000 T  0556
            BEGIN                                                                                   05102000 T  0556
                LABEL LOOP, XIT;                                                                    05103000 T  0556
                                                                                        START OF SEGMENT **********  146
            IF DEBUGTOG THEN FLAGROUTINE("  FORM","ALPP  ",TRUE ) ;                                05103010 T  0000
                    PARMS ← 0;                                                                      05104000 T  0002
                    SCAN;                                                                           05105000 T  0002
                    IF NEXT ≠ ID THEN BEGIN FLOG(105); GO TO XIT END;                               05106000 T  0003
                    IF CLASS = FUNID THEN                                                           05106100 T  0008
                    IF FUNVAR = 0 THEN                                                              05107000 T  0008
                    BEGIN                                                                           05107020 T  0010
                    IF TYPE > 0 THEN                                                                05107030 T  0010
                    IF FUNVAR ← GLOBALSEARCH(NAME) ≠ 0 THEN                                         05107040 T  0011
                    IF BOOLEAN((T ← GET(FUNVAR)).TYPEFIXED) AND TYPE ≠ T.SUBCLASS                   05107050 T  0013
                        THEN FLAG(31);                                                              05107060 T  0016
                    PUT(FUNVAR ← FNEXT,GET(FNEXT) & VARID[TOCLASS]);                                05107100 T  0018
                    END;                                                                            05107160 T  0021
                    FNEW ← NEED(NNEW ← NAME, CLASS);                                                05108000 T  0021
                    ENTERX(NAME,IF CLASS = FUNID THEN                                               05108100 T  0023
                        1&GET(FNEW)[15:15:9] ELSE 1&GET(FNEW)[15:15:5]);                            05108200 T  0024
                    SCAN;                                                                           05109000 T  0029
                    IF NEXT ≠ LPAREN THEN                                                           05110000 T  0030
                        IF PARMSREQ THEN FLOG(106) ELSE FLSE                                        05111000 T  0030
                BEGIN                                                                               05112000 T  0033
                LOOP;                                                                               05113000 T  0033
                    SCAN;                                                                           05114000 T  0034
                    IF NEXT = ID THEN PARMLINK[PARMS ← PARMS+1] ← FNEXT ELSE                        05115000 T  0034
                    IF NEXT=STAR AND CLASS≠FUNID THEN PARMLINK[PARMS←PARMS+1]←0ELSE                 05116000 T  0038
                        FLOG(107);                                                                  05117000 T  0042
                    IF XREF THEN ENTERX(NAME,IF NEXT = STAR THEN 0 ELSE                             05117100 T  0044
                        0&GET(FNEXT)[15:15:9]);                                                     05117150 T  0047
                    SCAN;                                                                           05118000 T  0049
                    IF NEXT = COMMA THEN GO TO LOOP;                                                05119000 T  0050
                    IF NEXT ≠ RPAREN THEN FLOG(108);                                                05120000 T  0051
                    SCAN;                                                                           05121000 T  0053
                END;                                                                                05122000 T  0053
                    IF NOT ERRORTOG THEN DECLAREPARMS(FNEW);                                        05123000 T  0053
                XIT;                                                                                05124000 T  0055
            IF DEBUGTOG THEN FLAGROUTINE("  FORM","ALPP  ",FALSE) ;                                 05124010 T  0056
            END FORMALPP;                                                                           05125000 T  0058
                                                                                          146 IS   61 LONG, NEXT SEG    6
            PROCEDURE ENDS; FORWARD;                                                                05125100 T  0556
PRT(763) = ENDS
            PROCEDURE FUNCTION ;                                                                    05126000 T  0556
PRT(764) = FUNCTION
            BEGIN                                                                                   05127000 T  0556
                REAL A,B,C,I;  LABEL FOUND ;                                                        05127100 T  0556
                                                                                        START OF SEGMENT **********  147
```

```
STACK(F+2) = A
STACK(F+3) = B
STACK(F+4) = C
STACK(F+5) = I
                        IF SPLINK NEQ 0 THEN BEGIN FLAG(5); ENDS; SEGMENTSTART; END;          05128000 P  0000
                        LABL ← BLANKS;                                                       05128100 T  0003
                        FORMALPP(TRUE, FUNID);                                               05129000 T  0003
                        GETALL(FNEW, INFA, INFB, INFC);                                      05130000 T  0004
                        B←NUMINTM1 ;                                                         05130100 T  0006
                        WHILE A+1<B DO                                                       05130110 T  0007
                             BEGIN                                                           05130115 T  0008
                             IF C←INT[I←REAL(BOOLEAN(A+B) AND BOOLEAN(1022))]=INFB           05130120 T  0008
                             THEN GO FOUND ;                                                 05130125 T  0011
                             IF INFB<C THEN B←I.[36:11] ELSE A←I.[36:11] ;                   05130130 T  0012
                             END ;                                                           05130135 T  0016
                        IF INFB=INT[I←(A+B)×2^-1] THEN GO FOUND ;                            05130140 T  0017
                        IF FALSE THEN                                                        05130145 T  0020
                   FOUND: IF BOOLEAN(INT[I+1].INTSEEN) THEN BEGIN XTA←INFB; FLAG(167)END;    05130150 T  0020
                        IF TYPE<0 THEN TYPE←INFA.SUBCLASS&1[2:47:1] ;                        05131000 T  0025
                        PUT(SPLINK ← FNEW, INFA & 1[TOTYPF] & TYPE[TOSUBCL]);                05132000 T  0029
                        PUT(FUNVAR, GET(FUNVAR) & VARID[TOCLASS] & 1[TOTYPF] &               05133000 T  0032
                             TYPE[TOSUBCL]);                                                 05134000 T  0035
                   END FUNCTION;                                                             05135000 T  0037
                                                                          147 IS   40 LONG, NEXT SEG    6
                   PROCEDURE STATEMENT; FORWARD;                                             05136000 T  0556
PRT(765) = STATEMENT
                   PROCEDURE ASSIGN;                                                         05137000 T  0556
PRT(766) = ASSIGN
                   BEGIN                                                                     05138000 T  0556
                        LABEL XIT;                                                           05138100 T  0556
                                                                          START OF SEGMENT **********  148
                        EODS←TRUE ;                                                          05138110 T  0000
                        EXECUTABLE;                                                          05139000 T  0000
                        SCAN;                                                                05140000 T  0001
                        IF NEXT ≠ NUM THEN BEGIN FLOG(109); GO TO XIT END;                   05141000 T  0001
                        IF XREF THEN ENTERX(FNEXT,0&LABELID[TOCLASS]);                       05141500 T  0004
                        EMITNUM(FNEXT);                                                      05142000 T  0007
                        SCAN;                                                                05143000 T  0007
                        IF NEXT ≠ ID THEN BEGIN FLOG(105); GO TO XIT END;                    05144000 T  0008
                        IF XREF THEN ENTERX(XTA,1&GET(FNEXT)[15:15:9]);                      05144100 T  0010
                        EMITN(FNEXT);                                                        05145000 T  0014
                        EMITO(STD);                                                          05146000 T  0014
                        SCAN;                                                                05147000 T  0015
                   XIT:                                                                      05147100 T  0016
                   END ASSIGN;                                                               05148000 T  0016
                                                                          148 IS   17 LONG, NEXT SEG    6
                   PROCEDURE BLOCKDATA;                                                      05149000 T  0556
PRT(767) = BLOCKDATA
                   BEGIN                                                                     05150000 T  0556
                        IF SPLINK NEQ 0 THEN BEGIN FLAG(5); ENDS; SEGMENTSTART; END;         05151000 P  0556
                        LABL ← BLANKS;                                                       05151100 T  0560
                        SCAN;                                                                05152000 T  0560
                        SPLINK ← -1;                                                         05153000 T  0561
                   END BLOCKDATA;                                                            05154000 T  0562
                   PROCEDURE CALL;                                                           05155000 T  0562
PRT(770) = CALL
                   BEGIN                                                                     05156000 T  0562
```

```
                        EODS←TRUE ;                                                                  05156010 T  0562
                        EXECUTABLE;                                                                  05157000 T  0563
                        SCAN;                                                                        05158000 T  0564
                        SUBREF;                                                                      05159000 T  0564
                    END CALL;                                                                        05160000 T  0565
                    PROCEDURE COMMON;                                                                05161000 T  0565
PRT(771) = COMMON
                    COMMENT  THIS PROCEDURE MAKES THE COM ENTRY FOR COMMON ITEMS AND SETS            05161100 T  0565
                               THE CE BIT IN BOTH THE COM AND INFO TABLES AND LINKS                  05161110 T  0565
                               THE HEADS OF CHAINS;                                                  05161120 T  0565
                    BEGIN                                                                            05162000 T  0565
                        LABEL LOOP, BLOCK;                                                           05163000 T  0565
                                                                                  START OF SEGMENT **********  149
                        TYPE ← "2;                                                                   05164000 T  0000
                        SCAN;                                                                        05165000 T  0001
                        IF NEXT = ID THEN                                                            05166000 T  0001
                    BEGIN                                                                            05167000 T  0002
                        Z ← NEED(",BLNK,", BLOCKID);                                                 05168000 T  0002
                        IF XREF THEN ENTERX("BLANK.",0&BLOCKID[TOCLASS]);                            05168100 T  0004
                        GO TO BLOCK;                                                                 05169000 T  0007
                    END;                                                                             05170000 T  0010
                    LOOP:                                                                            05171000 T  0010
                        IF NEXT ≠ SLASH THEN FLOG(110);                                              05172000 T  0010
                        SCAN;                                                                        05173000 T  0012
                        IF NEXT = SLASH THEN Z ← NEED(",BLNK,", BLOCKID) ELSE                        05174000 T  0012
                    BEGIN                                                                            05175000 T  0015
                        IF NEXT ≠ ID THEN FLOG(105) ELSE                                             05176000 T  0017
                %   FORCE UNIQUE NAME BY APPENDING 1 TO NAME (2ND CHAR OF WORD)                      05176500 T  0019
                        BEGIN Z ← NEED(NAME&"1"[6:42:6],BLOCKID);                                    05177000 T  0019
                            IF XREF THEN ENTERX(NAME,0&BLOCKID[TOCLASS]);                            05177100 T  0022
                            SCAN;                                                                    05177200 T  0024
                        END;                                                                         05178000 T  0025
                        IF NEXT ≠ SLASH THEN FLOG(110);                                              05179000 T  0025
                    END;                                                                             05180000 T  0027
                        SCAN;                                                                        05181000 T  0027
                    BLOCK;                                                                           05182000 T  0027
                        IF (T←GET(Z+2)).ADINFO = 0 THEN                                              05183000 T  0028
                    BEGIN                                                                            05184000 T  0030
                        IF NEXTCOM←NEXTCOM+1>SUPERMAXCOM THEN                                        05185000 T  0031
                           BEGIN ROOT←0;  FATAL(124) END                                             05186000 T  0033
                        ELSE ROOT←NEXTCOM ;                                                          05186100 T  0035
                        PUTC(ROOT,0&HEADER[TOCLASS]&1[TOCE]&ROOT[TOADDR]) ;                          05186200 T  0036.
                        BAPC(Z) ;                                                                    05187000 T  0040
                    END  ELSE                                                                        05188000 T  0041
                    BEGIN                                                                            05189000 T  0041
                        ROOT ← T.ADINFO;                                                             05190000 T  0041
                        COM[(T←GETC(ROOT).LASTC).IR.T.IC].LINK←NEXTCOM+1 ;                           05191000 T  0042
                        IF COM[PWROOT]<0 THEN FLAG(2) ;                                              05191100 T  0049
                    END;                                                                             05192000 T  0053
                        DIMENSION;                                                                   05193000 T  0053
                        BAPC(0&ENDCOM[TOCLASS]) ;                                                    05194000 T  0053
                        COM[PWROOT].LASTC←NEXTCOM ;                                                  05195000 T  0055
                        PUT(T←GETC(ROOT+1)+2,GET(T)&ROOT[TOADINFO]) ;                                05196000 T  0059
                        IF NEXT ≠ SEMI THEN GO TO LOOP;                                              05197000 T  0064
                END COMMON;                                                                          05198000 T  0065
                                                                                  149 IS   66 LONG, NEXT SEG    6
                PROCEDURE ENDS;                                                                      05211000 T  0565
```

```
                BEGIN                                                                                    05212000 T  0565
                      IF SPLINK=0 THEN FLAG(184) ELSE                                          %112-     05212005 C  0565
                      BEGIN                                                                    %112-     05212007 C  0568
                        EODS←FALSE ;                                                                     05212010 T  0568
                        IF LOGIFTOG THEN FLAG(101);                                                      05213000 T  0569
                        LABL ← BLANKS;                                                                   05213100 T  0571
                        IF SPLINK < 0 THEN EMITO(XIT) ELSE EMITPAIR(0, KOM);                             05214000 T  0572
                        SEGMENT((ADR+4) DIV 4, NSEG, TRUE, EDOC);                                        05215000 T  0575
                      END;                                                                       %112-     05216000 P  0578
                END ENDS;                                                                                05217000 T  0578
                PROCEDURE ENTRY;                                                                         05218000 T  0579
PRT(772) = ENTRY
                BEGIN                                                                                    05219000 T  0579
                      REAL SP;                                                                           05220000 T  0579
                                                                                        START OF SEGMENT **********  150
STACK(F+2) = SP
                      IF SPLINK = 0 THEN FLAG(111) ELSE                                                  05221000 T  0000
                      IF SPLINK = 1 THEN BEGIN ELX ← 0; FLAG(4) END;                                     05222000 T  0002
                      LABL ← BLANKS;                                                                     05222100 T  0005
                      ADJUST ;                                                                           05222500 T  0006
                      SP ← GET(SPLINK);                                                                  05223000 T  0006
                      FORMALPP( (T←SP.CLASS) = FUNID, T);                                                05224000 T  0007
                      GETALL(FNEW, INFA, INFB, INFC);                                                    05225000 T  0010
                      IF INFA.CLASS = FUNID THEN                                                         05226000 T  0011
                           PUT(FNEW, INFA & 1[TOTYPE] & (SP.SUBCLASS)[TOSUBCL]);                         05227000 T  0013
                      PUT(FNEW+2, INFC & (ADR+1)[TOBASE]);                                               05228000 T  0017
                END ENTRY;                                                                               05229000 T  0019
                                                                                          150 IS   22 LONG, NEXT SEG    6
                PROCEDURE EQUIVALENCE;                                                                   05230000 T  0579
PRT(773) = EQUIVALENCE
                COMMENT  THIS PROCEDURE MAKES THE COM ENTRY FOR EQUIV ITEMS AND SETS                     05230100 T  0579
                           THE EQ BIT IN BOTH THE COM AND INFO TABLES AND LINKS                          05230110 T  0579
                           THE HEADS OF CHAINS;                                                          05230120 T  0579
                BEGIN                                                                                    05231000 T  0579
                      REAL P, Q, R, S;                                                                   05232000 T  0579
                                                                                        START OF SEGMENT **********  151
STACK(F+2) = P
STACK(F+3) = Q
STACK(F+4) = R
STACK(F+5) = S
                      BOOLEAN FIRST,PCOMM;                                                               05232050 T  0000
STACK(F+6) = FIRST
STACK(F+7) = PCOMM
                      LABEL XIT;                                                                         05232100 T  0000
                      IF LOGIFTOG THEN FLAG(101);                                                        05233000 T  0000
                      LABL ← BLANKS;                                                                     05233100 T  0002
                      DO                                                                                 05234000 T  0002
                    BEGIN                                                                                05235000 T  0003
                      FIRST ← FALSE;                                                                     05235500 T  0003
                      SCAN;                                                                              05236000 T  0003
                      IF NEXT ≠ LPAREN THEN BEGIN FLOG(106); GO TO XIT END;                              05237000 T  0004
                      IF NEXTCOM←NEXTCOM+1>SUPERMAXCOM THEN                                              05238000 T  0006
                         BEGIN ROOT←0;  FATAL(124) END                                                   05238100 T  0008
                      ELSE ROOT←NEXTCOM ;                                                                05238200 T  0010
                      PUTC(ROOT,0&HEADER[TOCLASS]&ROOT[TOADDR]) ;                                        05238300 T  0011
                      BAPC(0);  Q←0 ;                                                                    05239000 T  0014
                      DO                                                                                 05240000 T  0016
```

```
BEGIN
     SCAN;
     IF NEXT ≠ ID THEN BEGIN FLOG(105); GO TO XIT END;
     IF XREF THEN ENTERX(NAME,ORGET(FNEXT)[15:15:9]);
     FX1 ← FNEXT;
     LENGTH ← 0;
     SCAN;
     IF NEXT = LPAREN THEN
BEGIN
     IF GET(FX1).CLASS ≠ ARRAYID THEN
     BEGIN XTA ← GET(FX1+1); FLOG(112) END;
     R ← 0; P ← 1;
     S ← GET(FX1+2).ADINFO;
     DO
BEGIN
     SCAN;
     IF NEXT ≠ NUM OR NUMTYPE ≠ INTYPE THEN FLAG(113);
     LENGTH ← LENGTH + P×(FNEXT-1);
     P ← P×EXTRAINFO[(S+R).IR.(S+R).IC] ;
     R ← R-1;
     SCAN;
END  UNTIL NEXT ≠ COMMA;
     IF NEXT ≠ RPAREN THEN BEGIN FLOG(108); GO TO XIT END;
     IF R≠-1 THEN IF R←R+GET(FX1+2).NEXTRA≠0 THEN
        BEGIN  XTA←GET(FX1+1);  FLAG(IF R>0 THEN 23 ELSE 24) END ;
     SCAN;
END;
     IF (INFA←GET(FX1)) < 0 THEN
     BEGIN XTA ← GET(FX1+1); FLAG(39) END ELSE
BEGIN
     IF INFA.SUBCLASS > LOGTYPE THEN LENGTH ← 2×LENGTH ;
     BAPC(INFA&FX1[TOLINK]&LENGTH[TORELADD]&1[TOEQ]&ROOT[TOLASTC]);
     IF(PCOMM←BOOLEAN(INFA.CE)) OR BOOLEAN(INFA.EQ) THEN
BEGIN
     IF FIRST AND PCOMM THEN BEGIN XTA←GET(FX1+1); FLAG(2) END
     ELSE IF NOT FIRST THEN FIRST ← PCOMM;
     PUT(FX1.INFA & 1[TOEQ]);
     COM[NEXTCOM.IR.NEXTCOM.IC].LASTC ← A ← INFA.ADDR;
     B←GETC(ROOT).ADDR ;
     SETLINK(A) ;
     IF NOT RINGCHECK THEN
BEGIN
     COM[PWROOT].ADDR←GETC(A).ADDR ;
     PUTC(A.GETC(A)&B[TOADDR]&7[TOSUBCL]) ;
END
END  ELSE
     PUT(FX1.INFA & 1[TOEQ] & ROOT[TOADDR]);
     IF LENGTH > Q THEN Q ← LENGTH;
     IF BOOLEAN(INFA.FORMAL) THEN
     BEGIN XTA ← GET(FX1+1); FLAG(11) END;
END;
END  UNTIL NEXT ≠ COMMA;
     IF NEXT ≠ RPAREN THEN BEGIN FLOG(108); GO TO XIT END;
     SCAN;
     PUTC(ROOT+1.Q) ;
     BAPC(0&ENDCOM[TOCLASS]) ;
     COM[PWROOT].LASTC←NEXTCOM ;
```

```
05241000 T  0017
05242000 T  0017
05243000 T  0017
05243200 T  0020
05244000 T  0023
05245000 T  0024
05246000 T  0024
05247000 T  0025
05248000 T  0026
05249000 T  0026
05250000 T  0028
05251000 T  0031
05252000 T  0032
05253000 T  0035
05254000 T  0035
05255000 T  0035
05256000 T  0035
05257000 T  0038
05258000 T  0040
05259000 T  0045
05260000 T  0046
05261000 T  0046
05262000 T  0048
05262200 T  0050
05262300 T  0055
05263000 T  0060
05264000 T  0060
05265000 T  0060
05266000 T  0062
05267000 T  0065
05267100 T  0066
05268000 T  0069
05269000 T  0073
05270000 T  0076
05270100 T  0076
05270200 T  0080
05270500 T  0082
05271000 T  0084
05272000 T  0089
05272050 T  0091
05272100 T  0092
05272200 T  0093
05273000 T  0093
05274000 T  0098
05274200 T  0102
05275000 T  0102
05276000 T  0102
05277000 T  0105
05278000 T  0107
05279000 T  0108
05280000 T  0111
05281000 T  0111
05282000 T  0112
05283000 T  0115
05284000 T  0115
05285000 T  0117
05286000 T  0119
```

```
                    END  UNTIL NEXT ≠ COMMA;                                                                    05287000 T  0123
                    XIT;                                                                                        05287100 T  0124
                END EQUIVALENCE;                                                                                05288000 T  0125
                                                                                                          151 IS  129 LONG, NEXT SEG    6
                PROCEDURE EXTERNAL;                                                                             05289000 T  0579
PRT(774) = EXTERNAL
                BEGIN                                                                                           05290000 T  0579
                    IF SPLINK < 0 THEN FLAG(12);                                                                05291000 T  0579
                    IF LOGIFTOG THEN FLAG(101);                                                                 05292000 T  0581
                    LABL ← BLANKS;                                                                              05292100 T  0583
                    DO                                                                                          05293000 T  0583
                BEGIN                                                                                           05294000 T  0584
                    SCAN;                                                                                       05295000 T  0584
                    IF NEXT ≠ ID THEN FLOG(105) ELSE                                                            05296000 T  0584
                    BEGIN T ← NEED(NAME,EXTID);                                                                 05297000 T  0586
                        IF XREF THEN ENTERX(NAME,0&GET(T)[15:15:9]);                                            05297300 T  0588
                        SCAN;                                                                                   05297500 T  0591
                    END;                                                                                        05297800 T  0592
                                                                                                                05298000 T  0592
                END  UNTIL NEXT ≠ COMMA;                                                                        05299000 T  0592
                END EXTERNAL;                                                                                   05300000 T  0593
                PROCEDURE CHAIN;                                                                                05300100 T  0593
PRT(775) = CHAIN
                BEGIN                                                                                           05300150 T  0593
                    LABEL  AGN, XIT;                                                                            05300160 T  0593
                                                                                                          START OF SEGMENT **********  152
                    REAL  T1;                                                                                   05300170 T  0000
STACK(F+2) = T1
                    DEFINE  FLG(FLG1) = BEGIN FLOG(FLG1); GO TO XIT END#;                                       05300180 T  0000
                    EXECUTABLE;                                                                                 05300182 T  0000
                    SCAN;                                                                                       05300184 T  0000
                    T1 ← 2;                                                                                     05300190 T  0001
                    IF FALSE  THEN                                                                              05300210 T  0001
                AGN: IF GLOBALNEXT ≠ COMMA THEN FLG(28);                                                        05300220 T  0002
                    SCAN;                                                                                       05300230 T  0005
                    IF EXPR(TRUE)  >  REALTYPE THEN FLG(102);                                                   05300240 T  0006
                    IF (T1 ← T1 - 1) ≠ 0 THEN GO TO AGN;                                                        05300250 T  0009
                    IF GLOBALNEXT  ≠  RPAREN THEN FLG(3);                                                       05300260 T  0011
                    EMITPAIR (37,KOM);                                                                          05300270 T  0013
                    SCAN;                                                                                       05300280 T  0014
                    IF GLOBALNEXT ≠ SEMI THEN FLOG(117);                                                        05300290 T  0015
                 XIT: WHILE GLOBALNEXT ≠ SEMI DO SCAN;                                                          05300300 T  0017
                END  CHAIN;                                                                                     05300310 T  0020
                                                                                                          152 IS   23 LONG, NEXT SEG    6
                PROCEDURE GOTOS;                                                                                05301000 T  0593
PRT(776) = GOTOS
                BEGIN    LABEL XIT;                                                                             05302000 T  0593
                                                                                                          START OF SEGMENT **********  153
                    REAL ASSIGNEDID;                                                                            05302100 T  0000
STACK(F+2) = ASSIGNEDID
                    EODS←TRUE ;                                                                                 05302110 T  0000
                    EXECUTABLE;                                                                                 05303000 T  0000
                    SCAN;                                                                                       05304000 T  0001
                    IF NEXT = NUM THEN                                                                          05305000 T  0001
                BEGIN                                                                                           05306000 T  0002
                    LABELBRANCH(NAME, FALSE);                                                                   05307000 T  0003
                    SCAN;                                                                                       05308000 T  0004
```

```
      GO TO XIT;                                                                  05309000 T  0004
END;                                                                              05310000 T  0005
      IF NEXT = ID THEN                                                           05311000 T  0005
BEGIN                                                                             05312000 T  0005
      ASSIGNEDID ← FNEXT;                                                         05313000 T  0006
      IF XREF THEN ENTERX(XTA,0&GET(FNEXT)[15:15:9]);                             05313200 T  0007
      SCAN;                                                                       05313300 T  0010
      IF NEXT ≠ COMMA THEN FLOG(114);                                             05313600 T  0010
      SCAN;                                                                       05314000 T  0012
      IF NEXT ≠ LPAREN THEN FLOG(106);                                            05314300 T  0013
      DO                                                                          05314600 T  0015
BEGIN                                                                             05315000 T  0016
      SCAN;                                                                       05315300 T  0016
      IF NEXT ≠ NUM THEN FLOG(109);                                               05315600 T  0016
      EMITV(ASSIGNEDID);                                                          05316000 T  0018
      EMITNUM(FNEXT);                                                             05316300 T  0019
      EMITO(NEQL);                                                                05316600 T  0020
      LABELBRANCH(NAME, TRUE);                                                    05317000 T  0020
      SCAN;                                                                       05317300 T  0021
END   UNTIL NEXT ≠ COMMA;                                                         05317600 T  0022
      IF NEXT ≠ RPAREN THEN FLOG(108);                                            05318000 T  0023
      SCAN;                                                                       05318200 T  0025
      EMITPAIR(1, SSN);    % CAUSE INVALID INDEX TERMINATION                     05318400 T  0026
      EMITDESCLIT(10);                                                            05318600 T  0027
      GO TO XIT;                                                                  05319000 T  0027
END;                                                                              05320000 T  0028
      IF NEXT ≠ LPAREN THEN FLOG(106);                                            05321000 T  0028
      P ← 0;                                                                      05322000 T  0030
      DO                                                                          05323000 T  0031
BEGIN                                                                             05324000 T  0031
      SCAN;                                                                       05325000 T  0031
      IF NEXT ≠ NUM THEN BEGIN FLOG(109); GO TO XIT END;                          05326000 T  0031
      LSTT[P←P+1] ← NAME;                                                         05327000 T  0034
      SCAN;                                                                       05328000 T  0036
END   UNTIL NEXT ≠ COMMA;                                                         05329000 T  0036
      IF NEXT ≠ RPAREN THEN BEGIN FLOG(108); GO TO XIT END;                       05330000 T  0038
      SCAN;                                                                       05331000 T  0040
      IF NEXT ≠ COMMA THEN BEGIN FLOG(114); GO TO XIT END;                        05332000 T  0041
      SCAN;                                                                       05333000 T  0043
      T ← P+1;              % DONT LET EXPR WIPE OUT LSTT                        05334000 T  0044
      IF EXPR(TRUE) > REALTYPE THEN FLOG(102);                                    05335000 T  0045
      EMITPAIR(JUNK, ISN);                                                        05336000 T  0047
      EMITPAIR(1,LESS);                                                           05337000 T  0048
      EMITOPDCLIT(JUNK);                                                          05338000 T  0049
      EMITPAIR(P,GRTR);                                                           05339000 T  0050
      EMITO(LOR);                                                                 05340000 T  0051
      EMITOPDCLIT(JUNK);                                                          05341000 T  0052
      EMITL(3);                                                                   05342000 T  0053
      EMITO(MUL);                                                                 05343000 T  0053
                                                                                  05344000 T  0054
IF ADR+3×P > 4085 THEN BEGIN ADR←ADR+1; SEGOVF END;                               05345000 T  0054
      EMITO(BFC);                                                                 05346000 T  0058
      EMITPAIR(1, SSN);                                                           05347000 T  0059
      EMITDESCLIT(10);                                                            05348000 T  0060
      FOR I ← 1 STEP 1 UNTIL P DO                                                 05349000 T  0061
BEGIN                                                                             05349100 T  0063
      J ← ADR; LABELBRANCH(LSTT[I], FALSE);                                       05349200 T  0063
```

```
                    IF ADR#J = 2 THEN EMITO(NOP);                                   05349300 T  0065
                END;                                                                05349400 T  0067
                XIT:                                                                05350000 T  0069
                    IT ← 0;                                                         05351000 T  0070
            END GOTOS;                                                              05352000 T  0070
                                                                                  153 IS   73 LONG, NEXT SEG    6
            PROCEDURE IFS;                                                          05353000 T  0593
PRT(777) = IFS
            BEGIN   REAL TYPE, LOGIFADR, SAVELABL;                                  05354000 T  0593
                                                                                  START OF SEGMENT **********  154
STACK(F+2) = TYPE
STACK(F+3) = LOGIFADR
STACK(F+4) = SAVELABL
                    EODS←TRUE ;                                                     05354010 T  0000
                    EXECUTABLE;                                                     05355000 T  0000
                    SCAN;                                                           05356000 T  0001
                    IF NEXT ≠ LPAREN THEN FLOG(106);                                05357000 T  0001
                    SCAN;                                                           05358000 T  0003
                    IF TYPE ← EXPR(TRUE) = COMPTYPE THEN FLAG(89);                  05359000 T  0004
                    IF NEXT ≠ RPAREN THEN FLOG(108);                                05360000 T  0007
                    IF TYPE = LOGTYPE THEN                                          05361000 T  0009
                BEGIN                                                               05362000 T  0010
                    EMITB(-1, TRUE);                                                05363000 T  0010
                    LOGIFADR ← LAX;                                                 05364000 T  0011
                    LOGIFTOG ← TRUE; EOSTOG ← TRUE;                                 05365000 T  0012
                    SAVELABL ← LABL; LABL ← BLANKS;                                 05365100 T  0015
                    STATEMENT;                                                      05366000 T  0016
                    LABL ← SAVELABL;                                                05366100 T  0017
                    LOGIFTOG ← FALSE; EOSTOG ← FALSE;                               05367000 T  0017
                    FIXB(LOGIFADR);                                                 05368000 T  0020
                END ELSE                                                            05369000 T  0021
                BEGIN                                                               05370000 T  0021
                    IF TYPE = DOUBTYPE THEN                                         05371000 T  0021
                    BEGIN EMITO(XCH); EMITO(DEL) END;                               05372000 T  0022
                    SCAN;                                                           05373000 T  0024
                    IF NEXT ≠ NUM THEN FLOG(109);                                   05374000 T  0024
                    FX1 ← FNEXT; NX1 ← NAME;                                        05375000 T  0026
                    SCAN;                                                           05376000 T  0028
                    IF NEXT ≠ COMMA THEN FLOG(114);                                 05377000 T  0028
                    SCAN;                                                           05378000 T  0030
                    IF NEXT ≠ NUM THEN FLOG(109);                                   05379000 T  0031
                    FX2 ← FNEXT; NX2 ← NAME;                                        05380000 T  0033
                    SCAN;                                                           05381000 T  0034
                    IF NEXT ≠ COMMA THEN FLOG(114);                                 05382000 T  0035
                    SCAN;                                                           05383000 T  0037
                    IF NEXT ≠ NUM THEN FLOG(109);                                   05384000 T  0037
                    FX3 ← FNEXT; NX3 ← NAME;                                        05385000 T  0039
                    SCAN;                                                           05386000 T  0041
                    IF FX2 = FX3 THEN                                               05387000 T  0041
                BEGIN                                                               05388000 T  0042
                    FMITPAIR(0,GEQL);                                               05389000 T  0043
                    LABELBRANCH(NX1, TRUE);                                         05390000 T  0044
                    LABELBRANCH(NX3, FALSE);                                        05391000 T  0045
                    IF XREF THEN ENTERX(NX2,0&LABELID[TOCLASS]);                    05391200 T  0046
                END ELSE                                                            05392000 T  0048
                    IF FX1 = FX3 THEN                                               05393000 T  0048
                BEGIN                                                               05394000 T  0050
```

```
                    EMITPAIR(0,NEQL);
                    LABELBRANCH(NX2, TRUE);
                    LABELBRANCH(NX1, FALSE);
                    IF XREF THEN ENTERX(NX3,0&LABELID[TOCLASS]);
                END ELSE
                    IF FX1 = FX2 THEN
                BEGIN
                    EMITPAIR(0,LEQL);
                    LABELBRANCH(NX3, TRUE);
                    LABELBRANCH(NX1, FALSE);
                    IF XREF THEN ENTERX(NX2,0&LABELID[TOCLASS]);
                END ELSE
                BEGIN
                    EMITO(DUP);
                    EMITPAIR(0,NEQL);
                    EMITB(-1, TRUE);
                    EMITPAIR(0,LESS);
                    LABELBRANCH(NX3, TRUE);
                    LABELBRANCH(NX1, FALSE);
                    FIXB(LAX);
                    EMITO(DEL);
                    LABELBRANCH(NX2, FALSE);
                END;
                END;
            END IFS;

            PROCEDURE NAMEL;
PRT(1000) = NAMEL
            BEGIN LABEL NIM,XIT,ELMNT,WRAP;
```

```
                    05395000 T  0050
                    05396000 T  0051
                    05397000 T  0052
                    05397200 T  0053
                    05398000 T  0056
                    05399000 T  0056
                    05400000 T  0057
                    05401000 T  0058
                    05402000 T  0059
                    05403000 T  0060
                    05403200 T  0061
                    05404000 T  0063
                    05405000 T  0063
                    05406000 T  0064
                    05407000 T  0065
                    05408000 T  0066
                    05409000 T  0067
                    05410000 T  0068
                    05411000 T  0069
                    05412000 T  0070
                    05413000 T  0071
                    05414000 T  0071
                    05415000 T  0072
                    05416000 T  0072
                    05417000 T  0072
 154 IS   75 LONG, NEXT SEG    6
                    05430000 T  0593

                    05431000 T  0593
START OF SEGMENT **********  155
```

```
               IF SPLINK < 0 THEN FLAG(12);
               IF LOGIFTOG THEN FLAG(101);
               LABL ← BLANKS;
          SCAN;  IF NEXT ≠ SLASH THEN FLOG(110);
NIM:      SCAN;  IF NEXT ≠ ID THEN BEGIN FLOG(105);  GO TO XIT END;
          IF J ← (INFA ← GET(LADR2 ← FNEXT)).CLASS = UNKNOWN THEN
          PUT(LADR2,INFA&NAMELISTITOCLASS])
          ELSE IF J ≠ NAMELIST THEN
          BEGIN XTA ← GET(LADR2 + 1);
               FLAG(20);
          END;
          LSTT[LSTS ← LADR1 ← 0] ← NAME;
          IF XREF THEN ENTERX(NAME,0&NAMELIST[TOCLASS]);
          SCAN;  IF NEXT ≠ SLASH THEN FLOG(110);
ELMNT:    SCAN; IF NEXT ≠ ID THEN BEGIN FLOG(105); GO TO XIT END;
          LADR1 ← LADR1 + 1;
          IF (T ← GET(FNEW ← GETSPACE(FNEXT)).CLASS) > VARID THEN FLAG(48);
          GETALL(FNEW,INFA,INFB,INFC);
          IF XREF THEN ENTERX(INFB,0&INFA[15:15:9]);
          IF LSTS ← LSTS+1 = LSTMAX THEN BEGIN FLOG(78); GO TO XIT END ELSE
          LSTT[LSTS] ← NAME&INFA.CLASNSUB[2:38:10]&0[8:47:1];
          IF T = ARRAYID THEN
          BEGIN J ← INFC.ADINFO;
               I ← INFC.NEXTRA;
               IF LSTS + I + 1 > LSTMAX THEN
               BEGIN FLOG(78);  GO TO XIT END;
               LSTT[LSTS ← LSTS + 1] ← 0&I[1:42:6]   %  # DIMENSIONS
```

```
                    05432000 T  0000
                    05433000 T  0002
                    05433100 T  0004
                    05434000 T  0004
                    05435000 T  0007
                    05436000 T  0011
                    05437000 T  0014
                    05438000 T  0016
                    05439000 T  0018
                    05440000 T  0020
                    05441000 T  0021
                    05442000 T  0021
                    05442500 T  0023
                    05443000 T  0026
                    05444000 T  0028
                    05445000 T  0032
                    05446000 T  0033
                    05447000 T  0037
                    05447500 T  0039
                    05448000 T  0042
                    05448500 T  0045
                    05449000 T  0049
                    05450000 T  0050
                    05451000 T  0052
                    05451100 T  0053
                    05451200 T  0055
                    05452000 T  0057
```

```
                                        &INFA.ADDR[7:37:11]   %  REL ADR          05453000 T  0059
                                        &INFC.BASE[18:33:15]   %  BASE           05454000 T  0061
                                        &INFC.SIZE[33:33:15];   %  SIZE           05455000 T  0062
                    FOR T ← J STEP -1 UNTIL J = I + 1 DO                          05456000 T  0063
                    LSTT[LSTS ← LSTS + 1] ← EXTRAINFO[T,IR,T,IC];                 05457000 T  0068
               END ELSE BEGIN LSTT[LSTS←LSTS+1]←0&(INFA.ADDR)[7:37:11];            05458000 T  0073
               IF BOOLEAN(INFA.CE) THEN LSTT[LSTS]←LSTT[LSTS]&INFC.BASE[18:33:15] 05458400 T  0077
                                        &INFC.SIZE[33:33:15] END;                 05458600 T  0080
               SCAN;  IF NEXT = COMMA THEN GO TO ELMNT;                           05459000 T  0082
               IF NEXT ≠ SEMI AND NEXT ≠ SLASH THEN FLOG(115);                    05460000 T  0084
               LSTT[LSTS + 1] ← 0;                                                05461000 T  0087
               LSTT[0].[2:10] ← LADR1;                                            05462000 T  0088
               PRTSAVER(LADR2,LSTS + 2,LSTT);                                     05463000 T  0091
               IF NEXT ≠ SEMI THEN GO TO NIM;                                     05464000 T  0093
          XIT:                                                                    05465000 T  0094
          END NAMFL;                                                              05466000 T  0095
                                                                  155 IS   96 LONG, NEXT SEG    6
          PROCEDURE PAUSE;                                                        05467000 T  0593
PRT(1001) = PAUSE
          IF DCINPUT THEN BEGIN XTA←"PAUSE ";  FLOG(151) END ELSE                 05467100 T  0593
          BEGIN                                                                   05468000 T  0596
               EODS←TRUE ;                                                        05468010 T  0599
               IF TSSEDITOG THEN TSSED("PAUSE ",2) ;                              05468100 T  0599
               EXECUTABLE;                                                        05469000 T  0602
               SCAN;                                                              05470000 T  0602
               IF NEXT = SEMI THEN EMITL(0) ELSE                                  05471000 T  0603
               IF NEXT = NUM THEN                                                 05472000 T  0605
          BEGIN                                                                   05473000 T  0607
               EMITNUM(NAME);                                                     05474000 T  0608
               SCAN;                                                              05475000 T  0609
          END;                                                                    05476000 T  0609
               EMITPAIR(33, KOM);                                                 05477000 T  0609
               EMITO(DEL);                                                        05477100 T  0610
          END PAUSE;                                                              05478000 T  0611
          PROCEDURE TYPIT(TYP,TMPNXT);  VALUE TYP;  REAL TYP,TMPNXT ;             05479000 T  0611
PRT(1002) = TYPIT
          BEGIN                                                                   05480000 T  0611
          TYPE←TYP;  SCAN ;                                                       05480010 T  0611
          IF NEXT=16 THEN BEGIN TMPNXT←16; FUNCTION END ELSE DIMENSION ;          05480020 T  0613
          END OF TYPIT ;                                                          05480040 T  0617
          DEFINE COMPLEX              =TYPIT(COMPTYPE,TEMPNEXT) #,                05481000 T  0617
                 LOGICAL              =TYPIT(LOGTYPE ,TEMPNEXT) #,                05482000 T  0617
                 DOUBLEPRECISION      =TYPIT(DOUBTYPE,TEMPNEXT) #,                05483000 T  0617
                 INTEGERS             =TYPIT(INTYPE  ,TEMPNEXT) #,                05484000 T  0617
                 REALS                =TYPIT(REALTYPE,TEMPNEXT) #;                05484500 T  0617
          PROCEDURE STOP;                                                         05485000 T  0617
PRT(1003) = STOP
          BEGIN                                                                   05486000 T  0617
               RETURNFOUND ← TRUE;                                                05486100 T  0617
               EODS←TRUE ;                                                        05486110 T  0619
               EXECUTABLE;                                                        05487000 T  0620
               COMMENT INITIAL SCAN ALREADY DONE;                                 05488000 T  0621
               EMITL(1);                                                          05489000 T  0621
               EMITPAIR(16, STD);                                                 05490000 T  0621
               EMITPAIR(10, KOM);                                                 05491000 T  0622
               EMITPAIR(5, KOM);                                                  05492000 T  0623
               WHILE NEXT ≠ SEMI DO SCAN;                                         05493000 T  0624
```

```
                    END STOP;                                                                  05494000 T  0627
                    PROCEDURE RETURN;                                                          05495000 T  0627
PRT(1004) = RETURN
                    BEGIN LABEL EXIT;                                                          05496000 T  0627
                                                                                    START OF SEGMENT **********  156
                          REAL T, XITCODE;                                                     05497000 T  0000
STACK(F+2) = T
STACK(F+3) = XITCODE
                          RETURNFOUND ← TRUE;                                                  05497100 T  0000
                          EODS←TRUE ;                                                          05497110 T  0001
                          EXECUTABLE;                                                          05498000 T  0002
                                                                                               05498100 T  0003
                                                                                               05498200 T  0003
                          SCAN;                                                                05499000 T  0003
                          IF SPLINK=0 OR SPLINK=1 THEN                                         05499100 T  0003
                               BEGIN XTA←"RETURN"; FLOG(153); GO EXIT END ;                     05499110 T  0005
                          IF NEXT = SEMI THEN                                                  05500000 T  0009
                       BEGIN                                                                   05501000 T  0009
                    % VOID                                                                     05502000 T  0010
                          IF (T ← GET(SPLINK)).CLASS = FUNID THEN                              05503000 T  0010
                       BEGIN                                                                   05504000 T  0012
                          EMITV(FUNVAR);                                                       05505000 T  0013
                          IF T.SUBCLASS > LOGTYPE THEN EMITPAIR(JUNK, STD);                    05506000 T  0013
                          XITCODE ← RTN;                                                       05507000 T  0016
                       END  ELSE XITCODE ← XIT;                                                05508000 T  0017
                       IF ADR ≥ 4077 THEN                                                      05509000 T  0018
                          BEGIN ADR ← ADR+1; SEGOVF END;                                       05510000 T  0019
                          EMITOPDCLIT(1538);        % F+2                                     05511000 T  0021
                          EMITPAIR(3, BFC);                                                    05512000 T  0022
                          EMITPAIR(10, KOM);                                                   05513000 T  0023
                          EMITO(XITCODE);                                                      05514000 T  0024
                          EMITOPDCLIT(16);                                                     05515000 T  0025
                          EMITPAIR(1, SUB);                                                    05516000 T  0025
                          EMITPAIR(16, STD);                                                   05517000 T  0026
                          EMITO(XITCODE);                                                      05518000 T  0027
                          GO TO EXIT;                                                          05519000 T  0028
                       END;                                                                    05520000 T  0031
                          IF LABELMOM = 0 THEN FLOG(145);                                      05520100 T  0031
                          IF EXPR(TRUE) > REALTYPE THEN FLAG(102);                             05521000 T  0033
                          IF EXPRESULT = NUMCLASS THEN                                         05521100 T  0035
                          BEGIN IF XREF THEN ENTFRX(EXPVALUE,0&LABELID[TOCLASS]);              05521200 T  0036
                                ADR ← ADR-1;EMITL(EXPVALUE-1)                                  05521400 T  0039
                          END ELSE                                                             05521600 T  0041
                          EMITPAIR(1, SUB);                                                    05522000 T  0042
                          EMITOPDCLIT(LABELMOM);                                               05523000 T  0043
                          EMITO(MKS);                                                          05524000 T  0044
                          EMITL(9);                                                            05525000 T  0045
                          EMITOPDCLIT(5);                                                      05526000 T  0045
                                                                                               05527000 T  0046
                       EXIT;                                                                   05528000 T  0046
                    END RETURN;                                                                05529000 T  0047
                                                                                    156 IS   50 LONG, NEXT SEG    6
                    PROCEDURE IMPLICIT ;                                                       05529100 T  0627
PRT(1005) = IMPLICIT
                    BEGIN                                                                      05529105 T  0627
                    REAL R1,R2,R3,R4 ;                                                         05529110 T  0627
                                                                                    START OF SEGMENT **********  157
```

```
STACK(F+2) = R1
STACK(F+3) = R2
STACK(F+4) = R3
STACK(F+5) = R4
                    LABEL R,A,X,L ;
                  IF NOT(LASTNEXT=42 OR LASTNEXT=1000 OR LASTNEXT=30                    %110-
                        OR LASTNEXT=16 OR LASTNEXT = 11)                                %110-
                    THEN BEGIN FLOG(181); FILETOG←TRUE; GO X END ;
               R:   EOSTOG←ERRORTOG←TRUE; FILETOG←FALSE ;
                    MOVEW(ACCUM[3],ACCUM[2],0,3); SCAN; ERRORTOG←FALSE; FILETOG←TRUE ;
                    IF R1←IF R3←NEXT=18 THEN INTID ELSE IF R3=26 THEN REALID ELSE 0&
                          (IF R3=10 THEN DOUBTYPE ELSE IF R3=19 THEN LOGTYPE ELSE IF R3=
                           6 THEN COMPTYPE ELSE 0)[TOSUBCL]=0 THEN
                                BEGIN FLOG(182); GO X END ;
                    SCN←2;  SCAN ;
                    IF NEXT=STAR THEN IF R3≠10 THEN
                         BEGIN SCAN ;
                         IF NEXT=NUM AND NUMTYPE=INTYPE THEN
                              BEGIN
                              IF FNEXT=4 THEN BEGIN IF R3=6 THEN FLAG(176); GO L END ;
                              IF FNEXT=8 THEN
                                   BEGIN
                                   IF R3=26 THEN R1←0&DOUBTYPE[TOSUBCL]
                                   ELSE IF R3≠6 THEN FLAG(177) ;
                                   GO L ;
                                   END ;
                              END ;
                         FLAG(IF R3=26 THEN 178 ELSE 177-REAL(R3=6)) ;
               L:        NCR←REAL(NCR,[30:3]≠0)+3"677777"+NCR;  SCN←1;  SCAN ;
                         END ;
                    IF NEXT≠LPAREN THEN BEGIN FLOG(106); GO X END ;
               A:   SCAN;  R4←ERRORCT ;
                    IF R2←NAME,[12:6]<17 OR (R2>25 AND R2<33) OR (R2>41 AND R2<50)
                       OR R2>57 OR NAME,[18:30]≠"      " THEN FLAG(179) ;
                    SCAN ;
                    IF NEXT≠MINUS THEN
                         BEGIN IF ERRORCT=R4 THEN TIPE[IF R2≠"0" THEN R2 ELSE 12]←R1 END
                    ELSE BEGIN
                         SCAN ;
                         IF R3←NAME,[12:6]<17 OR (R3>25 AND R3<33) OR (R3>41 AND R3<50)
                            OR R3>57 OR NAME,[18:30]≠"      " THEN FLAG(179) ;
                         IF R3 LEQ R2 THEN FLAG(180) ;
                         IF ERRORCT=R4 THEN FOR R2←R2 STEP 1 UNTIL R3 DO
                              BEGIN
                              IF R2>25 AND R2<33 THEN R2←33 ELSE IF R2>41 AND R2<50
                                                                 THEN R2←50 ;
                              TIPE[IF R2≠"0" THEN R2 ELSE 12]←R1 ;
                              END ;
                         SCAN ;
                         END ;
                    IF NEXT=COMMA THEN GO A ;
                    IF NEXT≠RPAREN THEN BEGIN FLOG(108); GO X END ;
                    SCAN;  IF NEXT=COMMA THEN GO R ;
                    IF NEXT≠SEMI THEN BEGIN FLOG(117); GO X END ;
                    IF SPLINK>1 THEN
                         BEGIN
                         IF BOOLEAN(TYPE,[2:1]) THEN IF GET(SPLINK),CLASS=FUNID THEN
```

```
05529120 T  0000
05529130 P  0000
05529131 C  0002
05529140 T  0004
05529210 T  0007
05529215 T  0010
05529220 T  0014
05529230 T  0018
05529240 T  0022
05529250 T  0026
05529260 T  0028
05529270 T  0029
05529280 T  0031
05529290 T  0032
05529300 T  0034
05529310 T  0035
05529320 T  0038
05529330 T  0039
05529340 T  0040
05529350 T  0042
05529360 T  0045
05529370 T  0046
05529380 T  0046
05529390 T  0046
05529400 T  0049
05529410 T  0054
05529420 T  0054
05529430 T  0058
05529440 T  0059
05529450 T  0064
05529460 T  0068
05529470 T  0069
05529475 T  0070
05529480 T  0074
05529490 T  0077
05529500 T  0077
05529510 T  0082
05529520 T  0087
05529530 T  0089
05529540 T  0092
05529550 T  0092
05529560 T  0096
05529570 T  0098
05529580 T  0101
05529590 T  0104
05529600 T  0104
05529610 T  0104
05529620 T  0105
05529630 T  0108
05529635 T  0110
05529640 T  0112
05529650 T  0113
05529660 T  0113
```

```
                         BEGIN                                                                  05529670 T  0116
                         INFO[SPLINK,IR,SPLINK,IC],SUBCLASS←R3←TIPE[IF R3←GET(                  05529680 T  0117
                              SPLINK+1),[12:6]≠"0" THEN R3 ELSE 12],SUBCLASS ;                  05529690 T  0120
                         INFO[FUNVAR,IR,FUNVAR,IC],SUBCLASS←R3 ;                                05529700 T  0126
                         END ;                                                                  05529710 T  0131
                    IF R1←GET(SPLINK+2)<0 THEN                                                  05529720 T  0131
                         FOR R2←R1,NEXTRA-1+R1←R1,ADINFO STEP -1 UNTIL R1 DO                    05529730 T  0133
                              IF R3←PARMLINK[R2-R1+1]≠0 THEN                                    05529740 T  0140
                                   BEGIN                                                        05529750 T  0142
                                   EXTRAINFO[R2,IR,R2,IC],SUBCLASS←R4←TIPE[IF R4←               05529760 T  0143
                                        GET(R3+1),[12:6]≠"0" THEN R4 ELSE 12]                   05529770 T  0145
                                        .SUBCLASS ;                                             05529780 T  0150
                                   INFO[R3,IR,R3,IC],SUBCLASS←R4 ;                              05529790 T  0152
                                   END ;                                                        05529800 T  0156
                    END ;                                                                       05529810 T  0157
               X:   WHILE NEXT≠SEMI DO SCAN; FILETOG←FALSE ;                                    05529820 T  0157
                    END OF IMPLICIT ;                                                           05529830 T  0161

                                                                                  157 IS  164 LONG, NEXT SEG    6
               PROCEDURE SUBROUTINE;                                                            05530000 T  0627
PRT(1006) = SUBROUTINE
               BEGIN                                                                            05531000 T  0627
                    IF SPLINK NEQ 0 THEN BEGIN FLAG(5); ENDS; SEGMENTSTART; END;                05532000 P  0627
                    LABL ← BLANKS;                                                              05532100 T  0631
                    FORMALPP(FALSE, SUBRID);                                                    05533000 T  0631
                    SPLINK ← FNEW;                                                              05534000 T  0632
               END SUBROUTINE;                                                                  05535000 T  0633
               PROCEDURE MEMHANDLER(N); VALUE N; REAL N ;                                       05535010 T  0633
PRT(1007) = MEMHANDLER
                    BEGIN                                                                       05535020 T  0633
                    REAL A ;                                                                    05535030 T  0633

                                                                                  START OF SEGMENT **********  158
STACK(F+2) = A
                    LABEL L1,L2,L3,XIT ;                                                        05535040 T  0000
                    IF DEBUGTOG THEN FLAGROUTINE(" MEMHA","NDLER ",TRUE) ;                      05535045 T  0000
                    IF N LEQ 2 THEN                                                             05535050 T  0002
                         BEGIN % FIXED=1, VARYING=2.                                            05535060 T  0002
                         N←IF N=1 THEN 6 ELSE 0 ;                                               05535070 T  0003
               L1:       SCAN ;                                                                05535080 T  0006
                         IF NEXT≠ID THEN BEGIN FLOG(105); GO XIT END ;                          05535090 T  0006
                         IF (A←GET(GETSPACE(FNEXT))).CLASS≠ARRAYID THEN                         05535100 T  0011
                              BEGIN FLOG(35); GO XIT END ;                                      05535110 T  0013
                         IF XREF THEN ENTERX(XTA,0&A[15:15:9]) ;                                05535120 T  0015
                         IF BOOLEAN(A,EQ) OR BOOLEAN(A,FORMAL) THEN FLAG(169)                   05535130 T  0018
                         ELSE BEGIN                                                             05535140 T  0020
                              EMITO(MKS); EMITPAIR(A,ADDR,LOD); EMITL(N) ;                      05535150 T  0021
                              EMITV(NEED(",MEMHR",INTRFUNID)) ;                                 05535160 T  0024
                              END ;                                                             05535170 T  0026
                         SCAN; IF NEXT=COMMA THEN GO L1 ;                                       05535180 T  0026
                         END                                                                    05535190 T  0028
                    ELSE IF N=3 THEN                                                            05535200 T  0028
                              BEGIN % AUXMEMED FUNCTION OR SUBROUTINE.                          05535210 T  0030
                              SCAN ;                                                            05535220 T  0031
                              IF NEXT≠ID THEN BEGIN FLOG(105); GO XIT END ;                     05535225 T  0031
                              IF GET(FNEXT+1)≠GET(SPLINK+1) THEN                                05535230 T  0034
                                   BEGIN FLOG(170); GO XIT END ;                                05535235 T  0037
                              PUT(SPLINK,GET(SPLINK)&1[TOADJ]) ;                                05535240 T  0038
                              IF XREF THEN ENTERX(XTA,0&GET(FNEXT)[15:15:9]); SCAN ;            05535250 T  0041
```

```
                    END
               ELSE BEGIN % RELEASE.
L2:                 SCAN ;
                    IF NEXT≠ID THEN BEGIN FLOG(105); GO XIT END ;
                    IF (A←GET(GETSPACE(FNEXT))).CLASS=ARRAYID THEN
                         BEGIN
                         IF BOOLEAN(A,EQ) OR BOOLEAN(A,FORMAL) THEN FLAG(169)
                         ELSE BEGIN
                              EMITO(MKS);  EMITPAIR(A,ADDR,LOD) ;
                              EMITPAIR(1,SSN) ;
                              EMITV(NEED(",MEMHR",INTRFUNID)) ;
                              END ;
L3:                      IF XREF THEN ENTERX(XTA,0&A[15:15:9]) ;
                         END
                    ELSE IF A.CLASS≥BLOCKID OR A.CLASS≤LABELID THEN
                              BEGIN FLOG(171); GO XIT END
                         ELSE BEGIN
                              EMITPAIR(A,ADDR,LOD);  EMITPAIR(38,KOM) ;
                              EMITO(DFL);  GO L3 ;
                              END ;
                    SCAN;  IF NEXT=COMMA THEN GO L2 ;
                    END ;
XIT:IF DEBUGTOG THEN FLAGROUTINE(" MEMHA","NDLER ",FALSE) ;
    END OF MEMHANDLER ;

PROCEDURE STATEMENT;
BEGIN LABEL DOL1, XIT;

     REAL TEMPNEXT ;
STACK(F+2) = TEMPNEXT
                    BOOLEAN ENDTOG;                                        %112-
STACK(F+3) = ENDTOG
                    DO SCAN UNTIL NEXT ≠ SEMI;
                    IF NEXT=ID THEN ASSIGNMENT ELSE IF NEXT LEQ RSH1 THEN
                    CASE(TEMPNEXT←NEXT) OF
                BEGIN
PRT(1010) = *CASE STATEMENT DESCRIPTOR*
                    FLOG(16);
                    ASSIGN;
                    IOCOMMAND(4);      %BACKSPACE
                    BLOCKDATA;
                    CALL;
                    COMMON;
                    COMPLEX;
                    BEGIN EXECUTABLE; SCAN END;       % CONTINUE
                    IOCOMMAND(7);    %  DATA
                    BEGIN SCAN; TYPE ← -1; DIMENSION END;
                    DOUBLEPRECISION;
                    BEGIN ENDS; ENDTOG:=TRUE; SCAN END;                    %112-
                    FILECONTROL(1);    %ENDFILE
                    ENTRY;
                    EQUIVALENCE;
                    EXTERNAL;
                    BEGIN TYPE ← -1; FUNCTION END;
                    GOTOS;
                    INTEGERS;
                    LOGICAL;
```

```
05535420 T  0045
05535430 T  0045
05535440 T  0045
05535450 T  0046
05535460 T  0049
05535470 T  0051
05535480 T  0052
05535490 T  0054
05535500 T  0055
05535510 T  0058
05535520 T  0059
05535530 T  0060
05535540 T  0060
05535550 T  0063
05535560 T  0063
05535570 T  0068
05535575 T  0070
05535580 T  0071
05535585 T  0073
05535590 T  0074
05535595 T  0074
05535600 T  0076
05535605 T  0076
05535610 T  0079
 158 IS    84 LONG, NEXT SEG     6
05536000 T  0633
05537000 T  0633
START OF SEGMENT **********  159
05537100 T  0000
05537200 C  0000
05538000 T  0000
05539000 T  0001
05540000 T  0004
05541000 T  0006
05542000 T  0006
05543000 T  0008
05544000 T  0009
05545000 T  0010
05546000 T  0011
05547000 T  0012
05548000 T  0013
05549000 T  0015
05550000 T  0016
05551000 T  0017
05552000 T  0020
05553000 P  0022
05554000 T  0024
05555000 T  0025
05556000 T  0026
05557000 T  0027
05558000 T  0028
05559000 T  0030
05560000 T  0031
05561000 T  0033
```

```
          NAMEL;                                                                                    05562000 T  0035
          PAUSE;                                                                                    05563000 T  0036
          IOCOMMAND(2);      %PRINT                                                                 05564000 T  0037
          ;                                                                                         05565000 T  0038
          IOCOMMAND(3);      %PUNCH                                                                 05566000 T  0038
          IOCOMMAND(0);      %READ                                                                  05567000 T  0040
          REALS;                                                                                    05568000 T  0041
          RETURN;                                                                                   05569000 T  0043
          FILECONTROL(0);    %REWIND                                                                05570000 T  0044
          BEGIN SCAN; STOP END;                                                                     05571000 T  0045
          SUBROUTINE;                                                                               05572000 T  0047
          IOCOMMAND(1);      %WRITE                                                                 05573000 T  0048
          FILECONTROL(7);    %CLOSE                                                                 05573100 T  0049
          FILECONTROL(6);    %LOCK                                                                  05573200 T  0051
          FILECONTROL(4);    %PURGE                                                                 05573300 T  0052
          IFS;                                                                                      05574000 T  0053
          FORMATER;                                                                                 05575000 T  0054
          CHAIN;                                                                                    05575100 T  0055
          MEMHANDLER(1) ; %FIXED                                                                    05576000 T  0056
          MEMHANDLER(2) ; %VARYING                                                                  05576100 T  0058
          MEMHANDLER(3) ; %AUXMEM FOR SUBPROGRAMS                                                   05576200 T  0059
          MEMHANDLER(4) ; %RELEASE                                                                  05577000 T  0061
          IMPLICIT ;                                                                                05577100 T  0062
END ELSE IF NEXT=EOF THEN GO XIT ELSE BEGIN NEXT←0; FLOG(16) END ;                                  05578000 T  0063
                                                                                  START OF SEGMENT **********  160
                                                                                   160 IS   44 LONG, NEXT SEG  159
          LASTNEXT.[33:15]←TEMPNEXT ;                                                               05578100 T  0067
          IF NOT ENDTOG THEN IF SPLINK=0 THEN SPLINK:=1;                          %112-             05579000 P  0068
          ENDTOG:=FALSE;                                                          %112-             05579100 C  0071
          IF LABL ≠ BLANKS THEN                                                                     05580000 T  0072
               BEGIN                                                                                05581000 T  0073
                    IF DT ≠ 0 THEN                                                                  05582000 T  0073
               BEGIN                                                                                05583000 T  0074
               DOL1: IF LABL = DOLAB[TEST ← DT] THEN                                                05584000 T  0075
               BEGIN                                                                                05585000 T  0076
                    EMITB(DOTEST[DT], FALSE);                                                       05586000 T  0077
                    FIXB(DOTEST[DT],ADDR);                                                          05587000 T  0078
                    IF DT ← DT-1 > 0 THEN GO TO DOL1;                                               05588000 T  0079
               END ELSE                                                                             05589000 T  0082
                    WHILE TEST ← TEST-1 > 0 DO                                                      05590000 T  0082
                    IF DOLAB[TEST] = LABL THEN FLAG(14);                                            05591000 T  0085
               END;                                                                                 05592000 T  0088
                    LABL ← BLANKS;                                                                  05592100 T  0088
               END;                                                                                 05593000 T  0088
                                                                                                    05594000 T  0088
                    IF NEXT ≠ SEMI THEN                                                             05595000 T  0088
               BEGIN                                                                                05596000 T  0089
                    FLAG(117);                                                                      05597000 T  0090
                    DO SCAN UNTIL NEXT=SEMI OR NEXT=EOF ;                                           05598000 T  0090
               END;                                                                                 05599000 T  0093
                    ERRORTOG ← FALSE;                                                               05600000 T  0093
                    EOSTOG ← TRUE;                                                                  05601000 T  0094
               XIT;                                                                                 05602000 T  0095
          END STATEMENT;                                                                            05603000 T  0096
                                                                                  159 IS   99 LONG, NEXT SEG    6
          BOOLEAN STREAM PROCEDURE FLAGLAST(BUFF,ERR) ;                                             05603010 T  0633
PRT(1011) = FLAGLAST
```

```
                    BEGIN                                                                                  05603020 T  0633
                    LOCAL A;   SI←ERR;  8(IF SC≠" " THEN JUMP OUT;SI←SI+1;TALLY←TALLY+1);                  05603030 T  0634
                    A←TALLY;   SI←LOC A;   SI←SI+7 ;                                                       05603040 T  0637
                    IF SC<"8" THEN                                                                         05603050 T  0637
                       BEGIN   TALLY←1;   FLAGLAST←TALLY ;                                                 05603060 T  0638
                       DI←BUFF;DS←46 LIT"LAST SYNTAX ERROR OCCURRED AT SEQUENCE NUMBER ";                  05603070 T  0639
                       DS←LIT"""";   SI←ERR;   DS←8 CHR;   DS←LIT"""";                                     05603080 T  0645
                        DS←32 LIT "                                  ";              %510-                 05603081 C  0647
                        DS←32 LIT "                                  ";              %510-                 05603082 C  0651
                       END                                                                                 05603090 T  0655
                    END FLAGLAST ;                                                                         05603100 T  0655
PRT(1012) = *LIST, LABEL, OR SEGMENT DESCRIPTOR*
                INTEGER PROCEDURE FIELD(X);   VALUE X;   INTEGER X;                                        05603110 T  0657
                FIELD←IF X<10 THEN 1 ELSE IF X<100 THEN 2 ELSE IF X<1000 THEN 3 ELSE IF                    05603120 T  0657
                X<10000 THEN 4 ELSE IF X<100000 THEN 5 ELSE IF X<1000000 THEN 6 ELSE 7;                    05603130 T  0663
                FORMAT FOC1(/ "NUMBER OF SYNTAX ERRORS DETECTED = ",I*,",",X*,                             05604000 T  0675
                                                                                  START OF SEGMENT **********  161
PRT(1013) = FOC1
                        "NUMBER OF SEQUENCE ERRORS DETECTED = ",I*,","),                                   05605000 T  0675
                    FOC2("PRT SIZE = ",I*,";  TOTAL SEGMENT SIZE = ",I*,                                   05606000 T  0675
                        " WORDS;  DISK SIZE = ",I*," SEGS;  NO. PRGM. SEGS = ",I*,                         05607000 T  0675
                        ","),                                                                              05607010 T  0675
                    FOC3("ESTIMATED CORE STORAGE REQUIREMENT = ",I*," WORDS;",                             05608000 T  0675
                        "  COMPILATION TIME = ",I*," MIN, ",I*," SECS;",                                   05608010 T  0675
                        "  NO. CARDS = ",I*,","),                                                          05608020 T  0675
                    FOC4("ESTIMATED CORE STORAGE REQUIREMENT = ",I*," WORDS;"                              05608030 T  0675
                        "  COMPILATION TIME = ",I*," SECS;  NO. CARDS = ",I*,","),                         05608040 T  0675
                    FOC5("NUMBER OF TSS WARNINGS DETECTED = ",I*,",") ;                                    05608050 T  0675
                                                                                  161 IS  106 LONG, NEXT SEG    6
                COMMENT  MAIN DRIVER FOR FORTRAN COMPILER BEGINS HERE;                                     05609000 T  0675
                RTI ← TIME(1);                                                                             05610000 T  0675
                INITIALIZATION;                                                                            05611000 T  0677
                        DO STATEMENT UNTIL NEXT = EOF;                                                     05612000 T  0677
                  IF NOT ENDSEGTOG THEN IF SPLINK NEQ 0                                    %112-           05612100 P  0679
                     THEN BEGIN XTA:=BLANKS; FLAG(5); ENDS END;                            %112-           05612200 C  0681
                        WRAPUP;                                                                            05613000 T  0684
                POSTWRAPUP;                                                                                05613900 T  0685
                IF TIMETOG THEN IF FIRSTCALL THEN DATIME;                                                  05614000 T  0685
                IF NOT FIRSTCALL THEN                                                                      05615000 T  0688
                   BEGIN                                                                                   05616000 T  0689
                   WRITE(RITE,EOC1,FIELD(ERRORCT),ERRORCT,IF SEQERRCT=0 THEN 99 ELSE                       05617000 P  0689
PRT(1014) = *LIST, LABEL, OR SEGMENT DESCRIPTOR*
                   5,FIELD(SEQERRCT-1),SEQERRCT-1) ;                                                       05618000 T  0700
                   IF WARNED AND NOT DCINPUT THEN WRITE(RITE,EOC5,FIELD(WARNCOUNT),                        05618100 T  0709
PRT(1015) = *LIST, LABEL, OR SEGMENT DESCRIPTOR*
                   WARNCOUNT) ;                                                                            05618110 T  0719
                   WRITE(RITE,EOC2,FIELD(PRTS),PRTS,FIELD(TSEGSZ),TSEGSZ,FIELD(DALOC-1),                   05619000 T  0723
PRT(1016) = *LIST, LABEL, OR SEGMENT DESCRIPTOR*
                   DALOC-1,FIELD(NXAVIL),NXAVIL) ;                                                         05619010 T  0736
                   IF C1←(TIME(1)-RTI)/60 > 59 THEN WRITE(RITE,EOC3,FIELD(64×ESTIMATE),                    05619020 T  0747
PRT(1017) = *LIST, LABEL, OR SEGMENT DESCRIPTOR*
                   64×ESTIMATE,FIELD(C1 DIV 60),C1 DIV 60,FIELD(C1 MOD 60),C1 MOD 60,                      05619030 T  0757
                   FIELD(CARDCOUNT-1),CARDCOUNT-1) ELSE WRITE(RITE,EOC4,FIELD(ESTIMATE                     05619040 T  0771
PRT(1020) = *LIST, LABEL, OR SEGMENT DESCRIPTOR*
                   ×64),ESTIMATE×64,FIELD(C1),C1,FIELD(CARDCOUNT-1),CARDCOUNT-1) ;                         05619045 T  0785
                   IF ERRORCT>0 THEN IF FLAGLAST(ERRORBUFF,LASTERR) THEN WRITE(RITE,15,                    05619050 P  0802
                   ERRORBUFF[*]) ;                                                                         05619100 T  0809
```

```
                    END ;                                                                                    05619200 T  0811
                  END INNER BLOCK;                                                                           05620000 T  0811
PRT(1021) = *SEGMENT DESCRIPTOR*

                                                                                                        6 IS  815 LONG, NEXT SEG    2
                  END.                                                                                      05621000 T  0089
                                                                                                        2 IS   92 LONG, NEXT SEG    1
PRT(431) = OUTPUT(W) INTRINSIC, SEGMENT NUMBER =  162.
PRT(5) = BLOCK CONTROL INTRINSIC, SEGMENT NUMBER =  163.
PRT(557) = INPUT(W) INTRINSIC, SEGMENT NUMBER =  164.
PRT(515) = SORT INTRINSIC, SEGMENT NUMBER =  165.
PRT(446) = GO TO SOLVER  INTRINSIC, SEGMENT NUMBER =  166.
PRT(14) = ALGOL WRITE   INTRINSIC, SEGMENT NUMBER =  167.
PRT(15) = ALGOL READ    INTRINSIC, SEGMENT NUMBER =  168.
PRT(16) = ALGOL SELECT  INTRINSIC, SEGMENT NUMBER =  169.
PRT(514) = MERGE INTRINSIC, SEGMENT NUMBER =  170.
PRT(360) = FILE ATTRBUTS INTRINSIC, SEGMENT NUMBER =  171.

                                                                                                        1 IS    2 LONG, NEXT SEG    0
                                                                                                      172 IS   69 LONG, NEXT SEG    0
NUMBER OF ERRORS DETECTED =   0.  COMPILATION TIME =  400 SECONDS.

PRT SIZE = 530; TOTAL SEGMENT SIZE = 14260 WORDS; DISK SIZE = 720 SEGS; NO. PGM. SEGS = 172

ESTIMATED CORE STORAGE REQUIRED = 19044 WORDS.

ESTIMATED AUXILIARY MEMORY REQUIRED =     0 WORDS.

NUMBER OF CARD-IMAGES PROCESSED =   8251.
```

```
A
  0002 *00151000*

        00650000  00725000 *00894000* 00896000  00897000  00898000 *00909000* 00910000  00915160  01129000  01248000
        01655000  02610000 *03162007* 03162014 *03162030* 03162060  03162120 *03164050* 03164100 *03219000* 03220000
        03222000 *03224000* 03226000 *03976000* 03977000 *03997000* 03998000  04000000 *04002000* 04004000 *04075000*
       *04082000* 04082100  04099000  04101000  04102000  04105000  04111000  04113000  04114050  04114520  04115000
        04126100 *04136000* 04139000  04151000 *04155000* 04157000  04158000 *04350000* 04354000 *04354000**04359000*
        04359000  04487050  05063300  05063450  05063670 *05088000* 05089050  05090000  05091000  05127100 *05271000*
        05272050  05273000  05274000  05529120  05535030  05603030

A
  0006 *00408000*


A
  0006 *00416000*


A
  0006 *00418000*


A
  0006 *00706600*
        00706640  00706650  00706660

A
  0006 *00422220*
        00422230

A
  0006 *00707000*
        00711000  00720000

A
  0006 *00422000*


A
```

```
  0006 *00422235*
         00422240  00422250

A
  0006 *01364010*
         01364030 *01364030* 01364030

A
  0006 *01520000*
         01528000  01537000  01538000 *01538000* 01539000  01540000  01543000  01545000  01551000

A
  0006 *02317000*
         02319000

A
  0006 *03621000*
         03626000

A
  0006 *05603030*
         05603040

A
  0023 *00650000*
        *00678000**00685000* 00694000 *00695000* 00696100  00700000  00702000  00702100  00705100

A
  0023 *00651000*
         00672000

A
  0034 *00915160*
         00915220  00915280  00915300  00915480  00915560  00915580  00915600  00915620  00915660  00915680  00915700
         00915720  00915760  00915780  00915800  00915820  00915960  00916000  00916020  00916040  00916080  00916160
         00916300  00916400

A
  0030 *00725000*
```

```
          00748000  00749000  00754000  00755000  00759000  00761000  00763000  00764000  00764020  00764200  00764280
          00768000  00769000  00770000  00771000  00773100

A
  0041 *01129000*

         *01130000* 01130500  01131000 *01134000* 01134000  01135000  01138000  01140000  01144000  01147000  01148000
          01149000  01153000  01154000  01156000  01158000  01159000  01162000  01163100  01164000  01165000  01169000
          01171000

A
  0046 *01248000*

          01249000  01250000

A
  0062 *01655000*

         *01673000* 01674000  01676000  01689000

A
  0071 *01964000*

          01966000  01993000

A
  0075 *02042000*

          02042000  02046000

A
  0085 *02610000*

          02613000  02614030

A
  0080 *02362855*

          02362855  02362860

A
  0084 *02523100*

          02523300

A
  0081 *02514000*

          02518000
```

```
A
  0132 *04487050*
       *04513000*

A
  0147 *05127100*
        05130110  05130120 *05130130* 05130140

A
  0158 *05535030*
       *05535100* 05535120  05535130  05535150 *05535460* 05535480  05535500  05535540  05535560  05535580

A
  0157 *05529120*
       *05529430* 05529610

ACCIDENT
  0130 *04406010*
       *04415050* 04415280

ACCIDENT
  0139 *04786000*
       *04921000* 04941000  04949000

ACCUM
  0002 *00186000*
        02195000  02200000  02511000  02558000  02559000  02560000  02560100  02561000  02562000  02571100  02591100
        02620200  02731000  02812000  02819000  02820000  02822000  02825000  02831000  02839000  02850000  02878000
        02883000  03376000  03378100  03437000  05025000  05038900  05529215

ACCUMSTOP
  0002 *00186100*
        02818020 *03378100*

ACR
  0002 *00176000*
       *02500000* 02502000  02506020 *02811000* 02814000  02818020
```

ACR
0081 *02448000*

02478000

ACRV
0085 *02608000*

02608000 02611000

ACRV
0085 *02615000*

02615000 02616000

ACRV
0081 *02448000*

02449000 02452000 02474000 02477000 02482000 02496000 02497000

ACR0
0002 *00175000*

02577000 02587000 02597000 02601000 02620000 02623100 02811000 02864000 *03377000*

ACR1
0002 *00189000*

02500000 02622100 02622200 02749000 02782000 *03378000*

ACTION
0140 *04788000*

04792000 *04798000*

ACTIONLABELS
0139 *04787000*

04812000 04862000 04906000 04914000

ADD
0002 *00318000*

00921000 03665100 03682000 03696000 03717000 03734000 03766000 03990490 04256000 04261000 04268000
04269000 04270000 04271000 04272000 04557000 04855000 05055000

ADDR
0002 *00253000*

00674000 00685000 00688000 00691950 00755000 00764200 00764280 00770000 00818000 00846000 00856000
00877000 00972000 00991000 01159000 *01163100**01164000**01169000* 01281000 01291000 01300000 01312000
01326000 01328000 01332000 01338000 01347000 01348000 01350000 01351000 01470000 01475000 01589010
01610000 01631000 01676000 01687000 01707000 01791000 01801000 03147000 03209000 03243000 03252000
03649300 03716000 03725000 03738000 03743000 03774000 03805000 03988000 04102000 04105000 04139000
04151000 04157000 04404570 04415170 04445100 04500320 04503010 04503055 04503190 04531000 04542000
04591000 04773110 04874000 04880000 04891000 04897000 04956200 04956550 05063400 05088000 05089000
05090000 05271000 05272000 05273000 05453000 05458000 05535150 05535500 05535580 05587000

ADINFO
0002 *00259000*

00712000 00819000 03155000 03650000 03858000 04462000 04503200 05183000 05190000 05252000 05450000
05529730

ADINFO
0006 *00706150*

00706220 00706260 00706280 00706360

ADJ
0002 *00249000*

00668000 01090000 01099000 *01113000*

ADJUST
0006 *00407000*

00925020 00925060 00931000 00952000 00961000 00971000 00979000 00987000 00994000 *01376000* 01392000
01409000 02000000 02007000 02094000 03989000 04415030 04415090 04415250 04521000 04526000 04548000
04816135 04920000 04920200 04923000 04977000 05041300 05041600 05047000 05222500

ADR
0002 *00166000*

00396000 00397000 00629000 *00782000* 00929000 *00930000* 00930000 00932700 00933000 00935000 00953000
00964000 00972000 00980000 00984000 00988000 00998000 01232000 01233000 01233100 01269000 *01269000*
01270000 01294000 *01295000* 01295000 01355000 *01355000* 01356000 01361000 *01361000* 01362000 01370000
01371000 *01371000* 01377000 01387000 *01388000* 01388000 01389000 01393000 *01393000* 01394000 *01395000*
01395000 01396000 01397000 *01397000* 01398000 01399000 *01399000* 01400000 01410000 *01411000* 01411000
01423000 01423000 01431000 *01431000* 01432000 01434000 01435000 01435010 01526000 01527000 *01527000*
01532000 01539000 01545000 01551000 01562000 01563000 *01563000* 01566000 *01566000* 01567000 01574000
01575000 *01575000**01578000* 01578000 01579000 01602000 *01602000* 01603000 01604000 *01604000**01607000*
01608000 01662500 01662700 *01662700* 01663000 01665000 *01665000* 01666000 01668000 *01676000* 01682000
01683000 01683000 01683100 *01685000**01695000* 01695100 01786000 *01787000* 01787000 01795000 01808000
01811000 02001000 02004000 *02004000* 02006000 *02007000* 02007000 *02007000* 02008000 *02011000* 02051000
02094000 *02278300* 02278300 02380230 02389500 02401230 02409500 02432200 02432300 03289000 03654000
03655500 *03659000**03664100* 03876500 *03876500* 03990315 *03990315* 03990542 *03990542* 04178500 *04350000*
04415040 *04415040* 04415040 04415050 04415090 04518100 *04518200* 04518200 04527000 04549000 04601000
04604010 *04604025* 04611000 *04612000* 04612000 04773060 04773140 04773160 04816136 04866000 04866100
04866100 04876000 *04887000* 04887000 04893000 04902000 04920100 04920200 *04920200* 04921000 04923000
04978000 05000000 05041400 05041700 05048000 05215000 05228000 05345000 *05345000* 05349200 05349300

```
        05509000  05510000 *05510000**05521400* 05521400

ADVANCE
  0081 *02448000*

        02485000  02486000  02502000  02814000

AD2
  0002 *00329000*

        04236000  04260000  04264000  04267000  04281000

AGN
  0152 *05300160*

       *05300220* 05300250

AIMAG
  0118 *03990040*

       *03990240* 03990440

AINT
  0118 *03990040*

       *03990250* 03990420  03990490  03990565

ALF
  0090 *02987000*

       *03176000**03181000* 03187100  03195000

ALPH
  0077 *02150000*

       *02220000* 02239000

ALPH
  0081 *02451000*

        02469000  02471000

APHASF
  0134 *04642500*

        04715000

ARRAYDEC
```

0006 *00414000*

00765000 *01704000* 01734000 03211000

ARRAYID
0002 *00276000*

00695000 00762000 00839000 00865000 00909000 01479000 02569000 03647000 03822000 03823000 03887000
03918000 03919000 04097000 04111000 04415150 04501000 04885000 04887100 05017020 05023000 05071300
05249000 05449000 05535100 05535460

ARRY
0032 *00802000*

00840000 00868000

ARRY
0121 *04022000*

04104000 *04135000*

ARY
0006 *01026000*

01046000

ARY
0006 *01625000*

01633000 01648000

ARY
0134 *04673500*

04674500

ARYSZ
0002 *00218000*

01727000 01727000 03359000

ASK
0134 *04656100*

04708510 04718100 *04722500* 04725550 04726510 *04726520* 04735700 *04735740* 04744100 *04744110**04764000*

ASSIGN

```
  0006 *05137000*
        05148000  05543000

ASSIGNEDID
  0153 *05302100*
       *05313000* 05316000

ASSIGNMENT
  0006 *05008000*
        05063000  05539000

B
  0002 *00151000*
        01071600  01406000  01585000  02610000 *03162007* 03162014  03162030 *03162060* 03162120 *03219000* 03220000
        03222000 *03224000* 03226000 *03976000* 03977000 *03997000* 03998000  04000000 *04002000* 04004000 *05089000*
        05091000  05127100 *05272000* 05274000

B
  0006 *00418000*

B
  0006 *00416000*

B
  0006 *00422000*

B
  0006 *00404000*

B
  0006 *00649000*
        00689100

B
  0006 *00422235*
```

```
          00422235  00422240  00422250

B
  0006 *00706525*
          00706530

B
  0006 *00706600*
          00706640  00706650  00706660

B
  0006 *01364010*
         *01364030* 01364030

B
  0006 *01187000*
          01188000

B
  0006 *02317000*
          02317000  02319000

B
  0006 *03620000*


B
  0040 *01071600*
          01091000  01096000

B
  0058 *01585000*
        *01589000**01589010**01590000* 01591000

B
  0052 *01406000*
          01426000 *01426000*

B
```

0084 *02523100*

02523300

B

0080 *02362010*

02362010 02362220

B

0080 *02343000*

02343000 02347000

B

0085 *02610000*

02614020

B

0080 *02337100*

02337100 02337200

B

0147 *05127100*

05130100 05130110 05130120 *05130130* 05130140

BACK

0140 *04789000*

04792000 04801000 04804000 *04805000*

BACKNCR

0134 *04676000*

04676500 04680000 04687650 04689000 04708530 04730140 04737000

BAPC

0006 *00646500*

05085000 05187000 05194000 05239000 05268000 05285000

BASE

0002 *00255000*

00669000 00773000 00773200 00914000 01091000 01099000 01100000 *01111000* 01280000 01287000 01311000

```
        01325000  01331000  01589000 *01589010**01590000* 01601000  01602000  01677000  01800000 *01801000* 02862500
        03596100  03714000  03987000  04100000 *04447000* 04773150  04773160  04875000  04879000  04892000  04896000
        04956100  05454000  05458400

BASFNSIZE
  0002 *00257000*

        04415220  04509000

BASS
  0040 *01071000*

        01079000  01089000  01099000  01100000  01101000

BBC
  0002 *00318000*

        01553000  03990560

BBW
  0002 *00318000*

        01553000

BCL
  0006 *00631100*

         00631100  00632000  00636000  00641000  00646100  00646300  00646500

BDLINK
  0028 *00709000*

       *00712000* 00715000  00717000 *00717000*

BDLINK
  0113 *03634000*

       *03650000**03679000* 03679000 *03688000* 03688000

BDLINK
  0132 *04487100*

       *04503200* 04509150 *04509150* 04509200

BDPRT
  0002 *00200000*

        00953000  03283000
```

```
BDX
  0002 *00200000*

       *00953000* 00953000  03282000

BEGINSUB
  0036 *00946000*

       *00980000* 00993000

BEGINSUB
  0141 *04963000*

       *04978000* 04999000

BEOREF
  0057 *01524000*

       *01539000* 01546000  01552000

BFC
  0002 *00318000*

        01534000  01554000  01794000  01804000  01810000  03990320  03990555  04404525  04615100  05346000  05512000

BFW
  0002 *00318000*

        01534000  01554000  01666000  01794000  01804000  01810000

BIGGESTFILENB
  0002 *00207000*

        02982000  03150000

BIGJ
  0072 *01996000*

        02005000 *02008000*

BK
  0081 *02802200*

       *02862200* 02939100

BK1
```

0081 *02451000*

02455000 02463000

BK2
0081 *02451000*

02456000 02470000

BLANKCARD
0076 *02077100*

02077700 02077800 02086000

BLANKIT
0006 *00422220*

00723650 00916260 01018320 02364360 02371006 02372060

BLANKOUT
0097 *03372000*

03374000 03374100 03374200

BLANKS
0002 *00178000*

00949200 01586000 01788000 01973000 01974500 01975000 02079000 02089000 02092000 02364130 02499000
02506030 02507000 02508000 02509000 02655500 02656100 02705000 02707500 02708100 02778100 02812000
02818030 03262000 *03563910* 03563920 *03568000* 04677500 04679500 04680700 04708530 04730140 04761000
04773040 04816100 04964200 05067100 05128100 05151100 05213100 05222100 05233100 05292100 05365100
05433100 05532100 05580000 05592100 05612200

BLKCNTRLINT
0002 *00355110*

00812000

BLOCK
0077 *02147000*

02231000 02232010 *02237012* 02237020 *02237020**02237040* 02237040 *02237050* 02238000

BLOCK
0149 *05163000*

05169000 *05182000*

BLOCKDATA

```
  0006 *05149000*
         05154000  05545000

BLOCKID
  0002 *00286000*
         03207000  03649300  04503190  05168000  05168100  05174000  05177000  05177100  05535560

BLOKF
  0093 *03100000*
         03209000

BOUND
  0028 *00709000*
        *00715000* 00716000

BOUND
  0113 *03636000*
        *03679000* 03680000 *03687000**03688000* 03691000  03692000  03698000  03699000

BOUNDS
  0006 *03788000*
        *03831000**03834500* 03841000  05072000

BRANCHES
  0002 *00212000*
         00789000  01531000  01532000  01693000  01697000  01998000  02014000  04531000

BRANCHLIT
  0006 *00927000*
         00936000  00992000  03987000  04415240  04519000

BRANCHX
  0002 *00212000*
        *00785000* 01530000 *01531000* 01531000  02014000 *02015000* 04531000 *04532000*

BUF
  0078 *02254600*
         02254600  02255200
```

```
BUF
  0078 *02245600*

        02245600  02246200


BUF
  0078 *02261200*

        02262700 *02263000**02263300* 02263300 *02263900* 02263900 *02264800* 02264800 *02267500* 02267500  02268400
       *02268400**02275160* 02275160 *02279300* 02279300 *02280300* 02280300  02281300 *02281300**02283700* 02283700
       *02284000* 02284000 *02288500* 02288500


BUF
  0078 *02249500*

        02249500  02250100


BUF
  0080 *02362900*

        02364380 *02364380* 02364382 *02364382**02364385* 02364385 *02364420* 02364420 *02364460* 02364460  02364500
       *02364500**02364580* 02364580 *02364620* 02364620 *02371002**02371004* 02371004 *02372040**02372050* 02372050
       *02382200* 02382300 *02382300**02404200* 02404300 *02404300**02414280* 02414290 *02414290**02419200* 02419250
       *02419250*


BUF
  0080 *02333100*

        02333120  02333210


BUF
  0080 *02338000*

        02340000


BUFF
  0002 *00198000*

        02145000  03379000


BUFF
  0006 *05603010*

        05603070  05603110


BUFF
  0077 *02145000*
```

02233000 02236000 02238000

BUFF
  0080 *02362300*

  02362450

BUFL
  0002 *00198000*

  02051000 02262800 02364240 02364830 02364850 02420070 02426040 *03379000*

BUILDFPB
  0090 *03041000*

  03052000 03054000 03203000

BUMPADR
  0006 *00396000*

  01269000 01355000 01361000 01393000 01395000 01397000 01399000 01423000 01431000 01566000 01578000

BUMPLOCALS
  0006 *00396500*

  00690000 01138000 01164000 01166000 03831000 04453000 04979000 05079000

BUMPPRT
  0006 *00396600*

  00689100 01058000 01163100 01165100 01169000 01590000 01636000 01639000 01644000 01657000 01662000
  01679000 01696000 02182000 04586500 04607000 04932000 04936000

B1
  0006 *01261110*

  01261130

B2D
  0006 *00404000*

  00771000 01065000 *01187000**01188000* 01209000 01220000 01232000 01233000 01434000 01435000 03204000
  03209000 03244000 03252000

C
  0006 *00422000*

```
C
  0006 *00416000*

C
  0006 *00408000*

C
  0006 *00419000*

C
  0006 *00401000*

C
  0006 *00707000*
        00711000  00712000  00721000

C
  0006 *01778000*
        01794000  01799000  01804000  01810000

C
  0006 *01520000*
        01534000  01547000  01548000  01553000  01554000

C
  0006 *02321000*
        02323000

C
  0006 *03573000*
        03580000  03582000  03586000  03587000  03611000  03612000  03614000  03615000  03616000

C
```

```
  0040 *01071900*
        01077000  01086000

C
  0059 *01600000*
       *01601000* 01602000

C
  0058 *01585000*
       *01589000* 01589010  01590000

C
  0080 *02362855*
        02362860

C
  0087 *02681000*
        02706000  02707000  02714000  02717000  02718000

C
  0080 *02337100*
        02337200

C
  0080 *02342010*
        02342010

C
  0087 *02683000*
        02698000

C
  0080 *02362020*
        02362030  02362033  02362210  02362220  02362275

C
  0118 *03990030*
       *03990070* 03990080 *03990098* 03990108  03990122  03990130  03990134  03990190  03990200
```

```
C
  0121 *04051000*
        04054000

C
  0147 *05127100*
       *05130120* 05130130

CA
  0077 *02140000*
       *02209000**02211000**02214500* 02232000

CALL
  0006 *05155000*
        05160000  05546000

CALLEQUIV
  0006 *00649000*
        00706000  01159000

CARD
  0006 *00385000*
        00391000  02373030  02384000  02390000  02406000  02410000  03383000

CARDCOUNT
  0002 *00147000*
        02377000 *02377000* 05619040  05619045

CASEL
  0081 *02802300*
        02862800

CASESTMT
  0081 *02803000*
       *02805000* 02832000  02858000  02965000

CASEO
```

```
  0081 *02803100*
       *02807999*

CASE1
  0081 *02803100*
        02808000 *02809000* 02835000

CASE10
  0081 *02803200*
       *02838999*

CASE11
  0081 *02803200*
       *02839999*

CASE12
  0081 *02803200*
       *02840999*

CASE13
  0081 *02803200*
       *02848999*

CASE14
  0081 *02803200*
       *02849999*

CASE2
  0081 *02803100*
        02822000 *02824000*

CASE3
  0081 *02803100*
        02821000 *02826000*

CASE4
  0081 *02803100*
       *02827999*
```

CASE5
0081 *02803100*

02808000 02819000 *02829000* 02835000

CASE6
0081 *02803100*

02830999

CASE7
0081 *02803100*

02833999

CASE8
0081 *02803200*

02836999

CASE9
0081 *02803200*

02837999

CB
0002 *00184000*

02373030 02380100 02380125 02384000 02387000 02390000 02401100 02401125 02406000 02407000 02410000
02411000 02416200 02417000 02491650 02492000 02492450 03383000 04667600 04668500 04669500

CD
0076 *02063000*

02065000

CD
0076 *02077100*

02077300

CD
0082 *02490400*

02490600 02496000 02497000

CD

```
  0082 *02489200*
         02489600

CD
  0082 *02487600*
         02488000

CD
  0136 *04665500*
         04666000

CD
  0136 *04660500*
         04661000

CD
  0136 *04663000*
         04664000

CDC
  0002 *00319000*
         01284000  03749000  03757000  03903150  03930200  04160000  04617000

CDIVBYD
  0121 *04028060*
         04320000 *04339010*

CE
  0002 *00243000*
         00680000  00743000  00763000  00773100  01077000  01156000  01277000  01308000  01448000  03713000  04099000
         04445100  05086000  05269000  05458400

CEBIT
  0024 *00654000*
        *00657000* 00672000

CHAIN
  0006 *05300100*
         05300310  05575100
```

```
CHAR
  0006 *02434000*

         02441000

CHAR
  0081 *02448000*

         02449000   02453000   02479000   02481000

CHAR
  0081 *02511000*

         02559000   02560000   02560100   02561000   02620200   02731000   02825000   02839000   02878000   02883000

CHAR
  0134 *04673500*

         04674000

CHCK
  0142 *05010500*

        *05015500*  05030000

CHECK
  0040 *01073000*

         01104000 *01115000*

CHECK
  0076 *02063000*

         02076000   02077000   02081000

CHECK
  0115 *03849000*

        *03856000*  03872000   03895000   03943000   03950000

CHECK
  0112 *03575000*

         03592000   03596100   03597000   03600000 *03608000*

CHECKDO
```

```
  0006 *01962000*
       *01987000* 01992000  05035000

CHECKEXP
  0081 *02514000*
       02519000  02521000  02944000

CHECKFUN
  0084 *02540000*
       02550000  02551000  02601000

CHECKINFO
  0006 *00724000*
       00777000  00925000  00955000

CHECKOCTAL
  0081 *02605000*
       *02624000* 02628000  02939000

CHECKPERIOD
  0081 *02726000*
       02800000  02879000  02913000

CHECKRESERVED
  0081 *02522000*
       02604000  02859000

CHECKTOG
  0002 *00169061*
       02269300  02285800  02420015  02420020  03384000

CHKEXP
  0088 *02729000*
       02731000 *02767000*

CHRO
  0002 *00175000*
       02814000 *03376000* 03377000
```

CHR1
0002 *00189000*

02502000 *03378000*

CHS
0002 *00319000*

03629000 03990340 04235000 04281000 04289007 04294000 04296000 04299000 04300000 04415260 04616000
04925000

CHUNK
0006 *00390000*

00392000 01002000 01003000 01036000 01037000 03072000 03073000 03336000 03337000

CIS
0002 *00358000*

00376000 00377000 00385000

CKDO
0071 *01964000*

01970000 01974000

CL
0006 *00723140*

00723150 00723240 01054000 01055000

CL
0032 *00801000*

00833000 00836000 00839000 00851000 00854100 00856100 00859000 00861000 00862000 00865000 00867000
00871000 00880000

CLASNSUB
0002 *00248000*

04415230 04517000 05448500

CLASS
0002 *00246000*

00664000 00706220 00706280 00723670 00737000 00739000 00739200 00741000 00749000 00769000 00806000
00823000 00833000 00835000 00867000 00909000 00916300 01018170 01081000 01094000 01095500 01109000

01109000 01121000 01130500 01131000 01165000 01277000 01307000 01446000 01479000 01588200 01673000
01791100 01988000 01990000 02569000 02862500 03144000 03207000 03213000 *03580000* 03585000 03596100
03598000 03610000 03611000 03615000 03647000 03803000 03822000 *03823000* 03872000 03880000 03897000
03904200 03918000 03922000 03927000 04082000 04111000 04113000 04114520 04115000 04415150 04452000
04465000 04498000 04500600 04679000 04773080 04871100 04885000 04997000 05016000 05023000 05063650
05071200 *05071300* 05078500 05224000 05226000 05249000 05436000 05446000 05503000 05529660 05535100
05535460 05535560

CLASS
0006 *05101000*

05106100 05108000 05108100 05116000

CLFSSC
0121 *04028050*

04271000 *04292010*

CLESSD
0121 *04028050*

04270000 *04291010*

CMP
0006 *00706600*

00706640 01018330

CMPA
0006 *00706560*

00706570 00706660 00915240 00915300 00915600 00915720 00915760 00915780 00915800 00916020 00916080

CMPARGT
0034 *00915220*

00915600 00915720 00915780 00916020 00916080

CMPARGT1?
0034 *00915220*

00915600 00915720 00915780 00916020 00916080

CMPARLS
0034 *00915280*

00915760 00915800

```
CMPARLS1?
  0034 *00915280*

        00915760  00915800


CMPLX
  0118 *03990040*

       *03990280* 03990340


CNSTSFENLAST
  0121 *04025500*

       *04072500**04178500* 04345000


CNTNAM
  0132 *04488100*

        04488500  04488600  04500750


CNTR
  0036 *00948000*

       *00999000* 01011000  01012000 *01013000* 01013000  01017000 *01021550* 01021600  01021700  01021750 *01021750*
        01021950

CNTR
  0037 *01028000*


COC
  0002 *00319000*

        01317000  01319000  01320000  03756000  04167000


CODE
  0006 *00392000*

        01012000  01015000  01021700  01021850  01046000  01047000  01193000  03077000  03080000  03343000  03345000
        03364000  03365000  04031000  04655500


CODE
  0043 *01193000*

       *01210000**01211000**01212000**01213000**01221000**01222000**01223000* 01229000  01233000


CODE
```

```
  0043 *01195000*
       *01204000*

CODE
  0121 *04031000*
        04209000  04231000  04232000  04245500  04247000  04252000  04253000  04273000  04274000  04302000  04303000
        04323000  04324000  04342000  04344000  04363000  04365000

CODE
  0134 *04655500*
        04688500 *04693200* 04709500  04710000 *04715000**04715600**04716000**04716500**04717000**04717500**04719000*
       *04721000* 04721500  04722000  04725500 *04729500* 04730100 *04730500**04731500* 04735600  04735720 *04740600*
        04744100  04744500 *04746050**04746750* 04756500  04757400  04757500  04758500  04760550  04760600  04760700
        04760940 *04760980* 04762500  04765500

CODETOG
  0002 *00168200*
        01065000  01271000  01357000  01363000  01401000  01425000  01433000  01568000  01580000  01662700  01664000
        01667000 *02269900**02272300**02276500*

COL1
  0086 *02631000*
       *02647000* 02648000  02666000

COM
  0002 *00215000*
        00646200  00646400  00646700  01087000  01089000  05063670  05088000  05090000  05191000  05191100  05195000
        05271000  05273000  05286000

COML
  0144 *05063620*
       *05063660* 05063670

COMM
  0070 *01824000*
        01848000  01872000  01881000  01886000  01893000 *01942000* 01949000

COMMA
  0002 *00303000*
        01856000  01933000  01943000  01948000  01949000  02153000  02961000  03668000  03819000  03947000  03981000
```

```
          03990170  04194000  04547000  04555000  04564000  04803000  04858000  04906000  04914000  04916000  04944000
          04972000  05028010  05029200  05030000  05049000  05097000  05098000  05119000  05261000  05281000  05287000
          05299000  05300220  05313600  05317600  05329000  05332000  05377000  05382000  05459000  05529610  05529630
          05535180  05535595

COMMAS
  0134 *04656600*

         *04687650**04730100* 04730100 *04730110* 04759100  04760700 *04760900*

COMMON
  0006 *05161000*

          05198000  05547000

COMNT
  0082 *02489200*

          02490000  02490200  02492450  02492550  02492600

COMNT
  0136 *04663000*

          04664500  04665000  04669500  04670100  04670500

COMPLETECHECK
  0084 *02523100*

          02523400  02523500  02571100  02591100

COMPLEX
  0006 *05481000*

          05548000

COMPTYPE
  0002 *00300000*

          01444000  01930000  04200000  04291015  04297000  04321000  04330000  04337000  04340000  04369000  04372000
          04993000  04993500  05067280  05067330  05067380  05359000  05481000  05529240  05548000

CON
  0088 *02730100*

         *02757100* 02797100

CONSTRUCT
  0113 *03635000*
```

```
          03728000  03741000 *03744000*

CONTIN
  0082 *02487600*

          02489000  02491700  02491800  02492000  02492200

CONTIN
  0136 *04660500*

          04662000  04662500  04668000  04668500  04668600

CONTINUE
  0081 *02487000*

         *02491400* 02494000  02503000  02658000  02709000  02815000

CONTINUE
  0134 *04659000*

         *04667000* 04673000  04769000

CONV
  0085 *02615000*

          02617000  02619000  02622200  02623100

CONVERT
  0006 *02434000*

          02440000  02445000  02752000  02783000  02866000  02869000

CORRECTINFO
  0023 *00651000*

          00675000  00697100  00704000

COUNT
  0085 *02608000*

          02614050  02614060  02620000  02622100

CPHASE
  0134 *04642600*

          04715600

CPLUS
```

0051 *01380000*

01389000

CPLUS
0052 *01408000*

01412000

CPP
0070 *01824000*

01916000

CR
0006 *00391000*

02373030 02384000 02390000 02406000 02410000 03383000

CRD
0002 *00185000*

00358000 00359000 00360000 00361000 00376000 00377000 00378000 00379000 00380000 00385000 00387000
00388000 00389000 02051000 02053000 02054000 02081000 02086000 02262800 02263000 02364240 02364830
02364850 02364860 02371002 02371006 02372040 02372060 02373045 02374053 02374060 02374065 02380100
02380125 02380200 02380230 02381000 02382200 02389400 02389500 02395000 02396100 02401100 02401125
02401200 02401230 02403000 02404200 02409400 02409500 02414200 02414260 02414280 02414382 02415000
02416100 02419000 02419100 02419200 02420011 02420020 02420040 02420050 02420057 02420060 02420070
02420120 02423000 02426000 02426005 02426030 02426040 03374000 03375000

CRn
0097 *03372000*

03373000 03573000 03620000 03621000 03632000 03633000 03788000 03842000 03969000 03990010 03991000
04020000 04051000 04404100 04404125 04429000 04486000

CTC
0002 *00330000*

CTF
0002 *00330000*

04150000 04415210 04508000

CTIMESR
0121 *04028060*

04319010 04338000 04339010

```
CTIMESR1
  0121 *04028060*

        04292010 *04319020*


CTIMESR2
  0121 *04028060*

        04289007 *04319030*


CTYP
  0070 *01817000*

       *01853000* 01858000  01879000  01883000  01892000  01897000 *01913400*


CTYP
  0121 *04028050*

       *04291015* 04318010  04319030


CUD
  0070 *01818000*

       *01853000* 01871000


CUR
  0070 *01817000*

       *01852000* 01870000  01878000  01883100  01892000  01898000 *01913300*


CURSEG
  0006 *00939000*

       *00999100* 01005000  01008000  01021450


C1
  0002 *00193000*

        02752000  02753000  02754000  02758000  02783000 *02786000* 02787000  02791000  02792000  02795000  02796000
        02797000 *02865000* 02866000  02869000  02870000  02874000 *02912000* 03990030 *04604030**05619020* 05619030
        05619045


C1
  0118 *03990030*

       *03990050**03990070* 03990070  03990082  03990084  03990094 *03990098* 03990117  03990160
```

```
C2
  0002 *00193000*

       *01018160* 01018180  01018190  02730050  03990030  04500655 *04812050*

C2
  0088 *02730050*

       *02753000* 02753000  02786000

C2
  0118 *03990030*

       *03990050* 03990070 *03990080* 03990082 *03990098**03990105* 03990108  03990130 *03990165**03990180**03990195*
        03990200

C3
  0118 *03990030*

       *03990050* 03990070  03990080  03990084  03990104  03990134  03990195  03990550  03990560

C4
  0118 *03990030*

       *03990098* 03990132 *03990160* 03990160

C5
  0118 *03990030*

       *03990140* 03990165

D
  0006 *00422000*


D
  0006 *00416000*


D
  0007 *00426000*

        00427000

D
  0006 *00647000*
```

```
       00647000  00648000  00649000

D
  0007 *00433100*
       00433200  00600020  00600030  00601000  00608000  00615000

D
  0023 *00650100*


D
  0058 *01585000*
       01589010 *01589010* 01589020

D
  0073 *02020000*
       02026000 *02026000**02027000* 02027000 *02028000* 02028000 *02028000* 02029000 *02029000* 02029000  02032000


D
  0086 *02638000*
       02641000

D
  0087 *02683000*
       02687000

D
  0086 *02633000*
       02635000  02636000

D
  0121 *04051000*
       04057000

D
  0134 *04655500*
      *04678000**04678500**04679000* 04679000  04772600
```

```
DAAT
  0139 *04784000*

        04817000 *04920000* 04945000

DALOC
  0006 *00393000*

        01002000 *01003000* 01007000  01015000 *01016000* 01016000  01021500  01021850  01021900 *01021900* 01036000
       *01037000* 01039000  01047000 *01048000* 01048000  03072000 *03073000* 03074000  03080000 *03081000* 03081000
        03336000 *03337000* 03338000  03345000 *03346000* 03346000 *03388000* 05619000  05619010

DATAB
  0139 *04786100*

       *04816150* 04946600

DATALINK
  0002 *00155100*

       *00784100* 00925030  04816137 *04946500*

DATAPRT
  0002 *00155100*

       *00784100* 00925010  00925020  00925040  01708100  01708300  04816130 *04816135* 04816140

DATASET
  0006 *01815000*

        01961000  04929000

DATASKP
  0002 *00155100*

       *00784100**00925030* 00925040

DATASTMTFLAG
  0002 *00169059*

        04500100  04785100  04815000  04928000  04957100

DATASTRT
  0002 *00155100*

       *00784100* 00925030 *04816136*

DATATOG
```

0002 *00169076*

01826000 01960000 02939000 04785100

DATATOG
0139 *04785100*

04815000 04928000 04957100

DATIMF
0006 *00407500*

01018130 *02018000* 02039000 02052000 02286700 02380220 02389300 02401220 02409300 02432000 05614000

DB
0002 *00184000*

02364780 02364800 02374470 02374500 02414200 02414370 02414380 02491700 02491800 02492550 04668000
04668600 04670100

DBLOW
0002 *00187000*

01853000 02758000 02759000 02796000 02797000 02797100 *02797300**02865000* 02874000 02875000 *02912000*
04180000

DCCONTIN
0082 *02490400*

02491000 02491200 02491650 02491700

DCCONTIN
0136 *04665500*

04666000 04666500 04667600 04668000

DCINPUT
0002 *00169084*

00598000 00598120 00604000 00611000 00739100 02217000 02380000 02380100 02401000 02401100 02414175
02491600 02492450 02492550 02655000 02656100 02707000 02708100 03187100 03385200 03385350 03385355
03385360 03894060 03894100 04406030 04415000 04604060 04667500 04669500 04670100 04767500 04768000
05467100 05618100

DCMOVF
0080 *02362855*

02380230 02389500 02401230 02409500

DCMOVFW
  0080 *02362010*

        02362280 02362290 02380125 02401125 02414200

DCPLACE
  0007 *00433100*

        00433500 00598040

DDT111
  0118 *03990040*

       03990370 03990410

DEBUG
  0006 *01189000*

        01234000 01271000 01357000 01363000 01568000 01580000 01662700 01664000 01667000

DEBUGADR
  0002 *00174310*

       00782000 01232000 01233100 *01233100* 01434000 01435010 *01435010*

DEBUGTOG
  0002 *00169000*

        00602000 00609000 00676010 00705110 00710010 00722010 00803010 00914010 00948010 01021959 01073010
        01124010 01706110 01733200 01825000 01959000 01996010 02016010 02176010 02243010 *02269900**02272300*
        02972000 03643000 03786000 03797000 03840000 03854000 03966000 03971010 03989010 03990047 03990815
        03995010 04018010 04061000 04403000 04406025 04427025 04431010 04484010 04491000 04578000 04583010
        04591010 04600010 04618010 04772605 04812010 04957110 05011000 05062010 05066010 05098010 05103010
        05124010 05535045 05535605

DEBUGWORD
  0006 *00405000*

       01235000 01251000 01401000 01425000

DECIMAL
  0134 *04655500*

        04708500 *04708500**04708550* 04708550 04709500 04710000 04710500 *04722500**04727500**04730137* 04733500
        04735715 *04735715* 04735720 04735730 *04735750* 04737600 04744100 *04744110**04757425**04757430* 04758000
       04758100 04759000 04760000 *04764500*

DECLAREPARMS

0006 *04429000*

04485000 05123000

DEL
0002 *00319000*

01455000 01472000 01487000 03138000 03990240 03990320 03990360 03990555 03990560 04056000 04057000
04248000 04289005 04291010 04318000 04339010 04358000 04367000 04368000 04369000 04370000 04371000
04372000 04373000 04374000 04375000 04404525 04604020 04993550 05372000 05413000 05477100 05535585

DESCRFQ
0002 *00169020*

01289000 *03782000**04092000**04415030**04415270**04492000**04543000**04920000**04927000*

DEST
0090 *03041000*

03043000 03045000

DEX
0006 *01599000*

01605000 01610000

DIA
0002 *00328000*

00764240 01364030 03624000 04516300

DIB
0002 *00328000*

00764240 01364030 04516300

DIG
0081 *02451000*

02462000 02464000

DIMENSION
0006 *05064000*

05099000 05193000 05480020 05551000

DIU

```
  0002 *00327000*
         04328000  04338000  04339000

DKAC
  0002 *00353000*
         01007000  01039000  03238000  03274000  03303000

DKADRC
  0090 *03008000*
         03329000

DKADRF
  0090 *03006000*


DKAI
  0002 *00352000*


DKAREAS7
  0002 *00208180*
         03161000

DLESSC
  0121 *04028050*
         04268000 *04289005*

DLESSC1
  0121 *04028060*
         04282010 *04289007*

DOITINLINE
  0006 *03990010*
        *03990088**03990104**03990130* 03990820  04114530  04115010

DOLAB
  0002 *00231000*
         01977000  05584000  05591000
```

```
DOLIST
  0002 *00169093*

        02270200  02285200  02289450  02364890

DOLLAR
  0002 *00303000*

        02943000  04735630

DOLLARS
  0134 *04656600*

       *04687650* 04735600 *04735610**04735620* 04759200  04760600 *04760900*

DOLOPT
  0006 *02245000*

        02289700  02383000  02405000  02414382

DOL1
  0159 *05537000*

       *05584000* 05588000

DONE
  0041 *01128000*

        01130000  01143000  01160000 *01172000*

DONE
  0084 *02523000*

        02578000 *02598000*

DOTEST
  0002 *00231000*

        05048000  05586000  05587000

DOUBLED
  0145 *05066005*

       *05068000**05072000* 05078500  05079800

DOUBLEPRECISION
```

0006 *05483000*

05552000

DOUBTYPF
0002 *00299000*

00711000 00720000 00773100 00854100 00856100 01165000 01314000 01323000 01335000 01455000 01461000
01472000 01474000 01484000 01493000 01858000 01869000 02733500 02749000 02757000 02779500 02780000
02789000 02873000 03703000 03747000 03772000 03830000 03903200 03904100 03928000 03930300 03930700
03934000 03936100 03943100 04123000 04158000 04179000 04260020 04296000 04368000 04993550 04993590
05067210 05067320 05069100 05371000 05483000 05529230 05529340 05552000

DPHASE
0134 *04643000*

04716000

DPP
0070 *01824000*

01860000

DT
0002 *00228000*

00786000 01977000 *01982000* 01982000 *05034000* 05034000 *05034000* 05041100 05048000 05582000 05584000
05586000 05587000 05588000 *05588000*

DT
0006 *01055000*

01060000 01066000

DTIMESC
0121 *04028050*

04311000 *04318010*

DTYP
0121 *04028050*

04260020 04329020

DTYPC
0002 *00343000*

01060000

```
DTYPF
  0002 *00342000*

       03318000

DUMMY
  0002 *00290100*

       02862500  03596100  03605000

DUMPSIZF
  0002 *00237000*

       00238000  01637000  01638000

DUP
  0002 *00320000*

       01316000  01339000  03665100  03750000  03775000  03990310  03990545  04161000  04353000  04354000  04404325
       04404500  04609000  05406000

DV2
  0002 *00329000*

       04329010  04333000  04336000

D2B
  0006 *00631100*

       00631100  00794400  02432300

E
  0006 *00399000*


E
  0006 *00415000*


E
  0007 *00426000*

       00426000  00432000

E
  0006 *01743000*
```

```
          01745000  01747000

E
  0006 *01768000*
          01771000  01774000

E
  0006 *01735000*
          01737000  01739000

E
  0006 *01758000*
          01761000  01764000

E
  0056 *01439000*
         *01442000* 01443000  01446000  01448000  01470000  01475000  01479000

E
  0080 *02362855*
          02362860

EDITCODE
  0139 *04786000*
         *04838000**04867000**04886000**04910000**04919000* 04919000  04950000

EDOC
  0002 *00180000*
          00629000  01260000  01270000  01356000  01362000  01394000  01396000  01398000  01400000  01423000  01432000
          01567000  01579000  01662500  01663000  01666000  01699000  03289000  05215000

EDOC
  0006 *00942000*
          01012000

EDOCI
  0006 *00397000*
          01270000  01356000  01362000  01394000  01396000  01398000  01400000  01423000  01432000  01567000  01579000
          01662500  01663000  01666000
```

ELBAT
0112 *03574000*

03580000 03584000 *03585000* 03587000 03593000 03601000 03602000 03606000 03607000 *03610000* 03612000
03613000 03616000

ELINK
0032 *00798000*

00805000 00815000 00914000

ELMNT
0155 *05431000*

05444000 05459000

ELX
0002 *00158000*

00764040 *00786000* 00969000 00985000 04435000 04444000 04446000 04452000 *04484000* 04484000 *04484000*
05222000

EMITB
0006 *00408000*

00914000 00925020 00925030 00956000 00966000 00993000 *01520000* 01557000 01708400 01799000 03988000
04415030 04525000 04540000 04546000 04551000 04552000 04559000 04816150 04920000 04946500 04976000
05041300 05041600 05044000 05046000 05058000 05363000 05408000 05586000

EMITD
0006 *00723100*

00764240 *01359000* 01364000 01364030 01364035 03624000 03625000 04516300

EMITDDT
0006 *01364010*

01364040 03990370 03990480 03990490

EMITDFSCLIT
0006 *00411000*

00764220 00809000 00829000 00876000 01281000 01300000 01312000 01338000 *01558000* 01569000 01592000
01710000 03168000 03187500 03193000 03626000 03665300 03725000 03738000 03774000 03990150 04143000
04147000 04415270 04503030 04509350 04604050 04841000 04845000 04847000 04926000 05318600 05348000

EMITL

0006 *00406000*

00764220 00810000 00811000 00812400 00828000 00920000 00922000 00924000 00925040 00935000 00954000
00981000 01280000 01283000 01291000 01311000 *01353000* 01358000 01367000 01383000 01390000 01461000
01462000 01474000 01501000 01510000 01511000 01545000 01551000 01591000 01605000 01712100 01713000
01714000 01719100 01720000 01720100 01721000 01722000 01724000 01725000 01731000 02006000 02008000
03132000 03133000 03134000 03138000 03139000 03163000 03165000 03166000 03169000 03170000 03184040
03187200 03187300 03187400 03187450 03187550 03187600 03187650 03187700 03187750 03188000 03189000
03190000 03191000 03194000 03195000 03196000 03197000 03198000 03200000 03201000 03665200 03707000
03724000 03727000 03737000 03740000 03760000 03889000 03903200 03923000 03928000 03930300 03930700
03932000 03934000 03936200 03943200 03990150 03990320 03990480 04054000 04103000 04142000 04146000
04248000 04358000 04363000 04404350 04404375 04404435 04404500 04404525 04404600 04404625 04409000
04415020 04415220 04415230 04415240 04415260 04415290 04503008 04510000 04516350 04517000 04518000
04520000 04555000 04584000 04609000 04615000 04616000 04617000 04619040 04619070 04832000 04833000
04848000 04856000 04863000 04863100 04877000 04879000 04883600 04894000 04896000 04903000 04910000
04912500 04917100 04925000 04939000 04942000 04950000 04952000 04953000 04956100 04956500 04956650
04956910 04956920 05049000 05342000 05471000 05489000 05521400 05525000 05535150

EMITLABFLDESC
0006 *01582000*

01596000 03868000 04952000 04953000 04956910 04956920

EMITLINK
0006 *01428000*

01533000 01793000 01809000 04875000 04892000 04902000

EMITN
0006 *00413000*

01273000 01302000 01453000 01458000 03782000 04092000 04516000 04545000 04558000 05035000 05041200
05056000 05145000

EMITNUM
0006 *00394000*

00716000 01287000 01325000 01327000 01331000 *01378000* 01403000 01408100 03162120 03171000 03172000
03173000 03660000 03680000 03696000 03707000 03712000 03733000 03765000 03768000 03884000 04100000
04180000 04415210 04415220 04506000 04509000 04509250 04509300 04516200 05142000 05316300 05474000

EMITNUM2
0006 *01404000*

01427000 04180000

EMITO
0006 *00409000*

00718000 00808000 00830000 00921000 00923000 00925020 00981000 00983000 *01267000* 01272000 01282000

01284000 01296000 01313000 01316000 01318000 01339000 01341000 01342000 01343000 01344000 01372000
01374000 01377000 01384000 01391000 01400600 01426600 01454000 01455000 01460000 01464000 01465000
01472000 01486000 01487000 01489000 01490000 01496000 01498000 01500000 01503000 01504000 01507000
01509000 01513000 01514000 01534000 01534100 01547000 01548000 01553000 01554000 01564000 01576000
01591000 01684100 01696100 01701000 01708300 01709000 01794000 01795100 01804000 01808000 01810000
01811100 02009000 03132000 03134000 03138000 03154000 03283000 03284000 03627000 03630000 03665100
03665400 03681000 03682000 03696000 03708000 03712000 03717000 03726000 03734000 03739000 03749000
03750000 03752000 03753000 03754000 03755000 03756000 03757000 03766000 03775000 03777000 03778000
03779000 03780000 03903150 03904100 03928000 03930200 03934000 03936400 03943300 03990045 03990230
03990240 03990260 03990270 03990280 03990290 03990310 03990320 03990340 03990360 03990390 03990420
03990430 03990440 03990480 03990490 03990545 03990550 03990555 03990560 04028020 04053000 04056000
04057000 04118000 04148000 04150000 04152000 04160000 04161000 04163000 04164000 04165000 04166000
04167000 04199000 04214000 04219000 04223000 04235000 04236000 04240000 04241000 04245300 04245400
04248000 04249000 04256000 04260010 04264000 04267000 04269000 04277000 04281000 04282010 04285000
04288000 04289005 04289007 04290000 04291010 04292010 04294000 04296000 04299000 04300000 04306000
04311000 04314000 04317000 04318000 04318010 04319010 04319020 04319030 04327000 04328000 04329010
04333000 04336000 04339010 04353000 04354000 04355000 04358000 04359000 04362000 04367000 04368000
04369000 04370000 04371000 04372000 04373000 04374000 04375000 04404325 04404350 04404425 04404450
04404500 04404525 04404625 04415011 04415080 04415210 04415220 04415230 04415240 04415250 04415260
04415270 04415280 04418000 04503020 04503040 04503050 04508000 04509400 04510000 04516350 04518000
04520000 04521000 04545000 04550000 04557000 04558000 04604020 04609000 04615100 04616000 04617000
04619030 04816140 04820000 04877000 04894000 04903000 04910000 04912200 04922000 04925000 04926000
04939000 04941000 04949000 04993550 04995000 05036000 05041200 05043000 05055000 05057000 05146000
05214000 05316600 05340000 05343000 05346000 05349300 05372000 05406000 05413000 05477100 05514000
05518000 05524000 05535150 05535500 05535585

EMITOPDCLIT
0006 *00410000*

00716000 00812000 00827000 00847000 00856200 00875000 00919000 00925020 00932700 00933000 00983000
01326000 01328000 01332000 01347000 01348000 01350000 01351000 01389000 01412000 01497000 01543000
01570000 01581000 01593000 01684000 01696000 01708300 01726000 01803000 03138000 03202000 03283000
03285000 03628000 03680000 03716000 03990045 03990490 03990545 03990555 04028020 04102000 04124000
04235000 04245300 04245400 04260010 04289007 04318010 04319020 04329010 04404325 04415070 04415200
04415210 04415280 04505200 04507000 04509200 04516200 04608000 04616000 04619070 04816140 04910000
04922000 04941000 04949000 04994000 05338000 05341000 05511000 05515000 05523000 05526000

EMITPAIR
0006 *00412000*

00720000 00721000 00764260 00764280 00812400 00845000 00846000 00848000 00855000 00856000 00856300
00877000 00925040 01317000 01319000 01320000 01340000 *01365000* 01375000 01413000 01452000 01459000
01463000 01469000 01470000 01475000 01483000 01495000 01499000 01502000 01508000 01512000 01606000
01732000 03139000 03623000 03629000 03661000 03665100 03751000 03776000 03888000 03990045 03990250
03990310 03990320 03990450 03990480 03990545 03990555 03990560 04028020 04105000 04138000 04151000
04162000 04235000 04240000 04245300 04245400 04260010 04263000 04282010 04284000 04289005 04292010
04311000 04313000 04318000 04319010 04329010 04332000 04404570 04410000 04411000 04412000 04415021
04415170 04415280 04503010 04503055 04606000 04848000 04855000 04856000 04863000 04880000 04897000
04939000 04941000 04949000 04956200 04956550 04979000 04993560 04993590 05214000 05300270 05318400
05336000 05337000 05339000 05347000 05389000 05395000 05401000 05407000 05409000 05477000 05490000
05491000 05492000 05506000 05512000 05513000 05516000 05517000 05522000 05535150 05535500 05535510
05535580

EMITSTORE

0006 *01437000*

01519000 04986000 05061000

EMITV
0006 *01303000*

01352000 01594000 03781000 04055000 04085000 04094000 04122000 04169000 04364000 04415300 04426000
04557000 04834000 04943000 04954000 04954050 04954100 04954150 04954200 04956700 04956750 04956800
04956925 04956930 05041400 05054000 05316000 05505000 05535160 05535520

ENDCOM
0002 *00294000*

00664000 01081000 01094000 01121000 05063650 05194000 05285000

ENDDEF
0090 *02988990*

ENDER
0134 *04657000*

04689000 04693500 04737500 04744505 *04756500*

ENDEXTCONDEF
0002 *00355990*

ENDFILDEF
0002 *00208490*

ENDP
0086 *02631000*

02648000 02655000 02657000 02663000 02666000

ENDP
0080 *02363000*

02371010 02372000 02374000 02374520 02391000 02397000 02412000 02414390 *02418000*

ENDPA
0080 *02362950*

```
          02374200  02420011 *02420550*

ENDPB
  0080 *02362902*

          02425100 *02426008*

ENDS
  0006 *05125100*

          05128000  05151000 *05211000* 05217000  05532000  05553000  05612200

ENDSAVTOGDFF
  0002 *00169199*



ENDSEGTOG
  0002 *00169073*

          00794000  01020000  02086000  03569000  05612100

ENDSUB
  0036 *00946000*
          *00984000* 00992000

ENDTOG
  0159 *05537200*

          *05553000* 05579000 *05579100*

ENDWRAPUP
  0090 *02997000*

           03290000 *03366000*

ENTER
  0006 *00415000*

          00695000  01588100 *01758000**01762000* 01767000  01811000  01985000  02862400  03584000  04493200  04678500
          04773060  04871000  04883350  04986000  05041100

ENTERING
  0006 *00600040*

          00600050
```

ENTERX
0006 *00706120*

00706500 01588400 01812100 01977100 01987100 02188100 03990134 04082100 04114050 04117100 04126100
04415165 04417100 04500655 04535500 04604040 04679400 04773175 04800100 04873000 04965100 04987100
05017055 05025200 05029400 05071400 05108100 05117100 05141500 05144100 05168100 05177100 05243200
05297300 05313200 05391200 05397200 05403200 05442500 05447500 05521200 05535120 05535250 05535540

ENTRY
0006 *05218000*

05229000 05555000

ENTRYLINK
0002 *00157000*

00804000 00989000 00995500 04435000 04452000

E0
0118 *03990045*

03990230 03990240 03990260 03990270 03990280 03990290 03990310 03990320 03990340 03990360 03990390
03990420 03990430 03990440 03990480 03990545 03990550 03990555 03990560

E0
0121 *04028020*

04028030 04028040 04235000 04236000 04245300 04245400 04260010 04281000 04282010 04289005 04289007
04291010 04292010 04311000 04318000 04318010 04319010 04319020 04319030 04329010 04339010

E0C1
0161 *05604000*

05617000

E0C2
0161 *05606000*

05619000

E0C3
0161 *05608000*

05619020

E0C4
0161 *05608030*

05619040

```
E0C5
  0161 *05608050*

        05618100

E0D5
  0002 *00168500*

        00706735 *01018110* 01018150  01018330  01018370 *04619025**04812020* 05017000 *05017060**05022010**05138110*
       *05156010**05212010**05302110**05354010**05468010**05486110**05497110*

E0F
  0002 *00301000*

        00706733  01018110  02830000  04788000  05578000  05598000  05612000

E0F
  0006 *00723450*

        00723520

E0F
  0027 *00706733*
        00706735 *00706760*

E0F
  0140 *04788000*
       *04791000* 04809000

E0F1
  0081 *02497000*
        02507000

E0F2
  0081 *02497000*
        02507000

E0L
  0118 *03990045*
        03990490  03990545  03990555

E0L
```

0121 *04028020*

04235000 04245300 04245400 04260010 04289007 04318010 04319020 04329010

E0STOG
0002 *00168000*

*00626000**02603000* 02827100 *02830000**02834000* 02855000 *05365000**05367000**05529210**05601000*

E0S1
0081 *02497000*

02508000

E0S2
0081 *02497000*

02508000

EP
0118 *03990045*

03990250 03990310 03990320 03990450 03990480 03990545 03990555 03990560

EP
0121 *04028020*

04028030 04028040 04235000 04245300 04245400 04260010 04282010 04289005 04292010 04311000 04318000
04319010 04329010

EPHASE
0134 *04643500*

04716500

EQ
0002 *00246100*

01156000 04500420 05086100 05269000 05535130 05535480

EQUAL
0002 *00303000*

02167000 02191000 02206000 02230090 02964000 04076100 04114200 04124100 04129100 04184100 04497000
04530000 04799000 04852000 04975000 05027000

EQUIV
0006 *00403000*

00691900 00692000 *01069000* 01122600 01125000

EQUIVALENCE
  0006 *05230000*

05288000 05556000

EQUL
  0002 *00320000*

04226000 04245000 04245500 04404500 04609000

EQVID
  0002 *00165000*

00696000 00696000 *03563940* 03563940 *03566000*

ERR
  0006 *05603010*

05603030 05603080

ERR
  0140 *04788000*

04793000 04806000

ERRMESS
  0006 *00423000*

00600000 00606000 00613000

ERROR
  0070 *01824000*

01837000 01843000 01863000 01868000 01888000 01896000 01897000 01899000 01914000 01920000 01928000
01929000 01933000 01937000 01940000 *01946000*

ERROR
  0132 *04488000*

04533000 04547000 04561000 04566000 *04569000* 04577000

ERROR
  0145 *05066000*

05068100 *05096100*

```
ERRORBUFF
  0002 *00169300*

       00598040  00598080  00598160  00598200  02054010  02420040  05619050  05619100


ERRORCT
  0002 *00145000*

       00434000 *00434000* 00598160  00606500  00613500  01009000  01043000  02281600  03290000  03365000  03385100
       03385200 *03385300* 05529430  05529475  05529530  05617000  05619050


ERRORTOG
  0002 *00167000*

       00602000  00609000 *00610000**00620000* 01440000 *01441000* 01518000 *01518000**02176100* 04121000  04129000
      *04574000**04760900* 05096000  05123000 *05529210**05529215**05600000*


ESTIMATE
  0002 *00147000*

       03164050 *03164050**03355000* 03356100  03361100  05619020  05619030  05619040  05619045


ES1
  0121 *04028030*

       04282010  04289005  04292010


ES11?
  0121 *04028030*

       04282010  04289005  04292010


ES2
  0121 *04028040*

       04311000  04318000


EUNF
  0002 *00208150*

      *02230050*


EWORD
  0032 *00803000*

      *00804000* 00805000  00806000  00818000


EX
```

```
  0115 *03847000*

       *03858000* 03872000  03880000  03897000  03900000  03904200  03913000  03918000  03922000  03927000  03937000
        03940000 *03944000* 03944000


EXACCUM
  0002 *00186000*

        02499000  02507000  02508000  02509000  02733000  02740000  02743000  02746000  02763000  02773000  02778000
        02837000  02838000  02839000  02840000  02841000  02842000  02914000  02917000  03378000  03378100  03437000


EXACCUMSTOP
  0002 *00186100*

        02506020 *03378100*


EXECUTABLE
  0006 *00403100*

        04619027 *04773010* 04773190  04819000  05017100  05022100  05139000  05157000  05300182  05303000  05355000
        05469000  05487000  05498000  05549000


EXIT
  0156 *05496000*
        05499110  05519000 *05528000*


EXP
  0006 *01437000*

        01443000  01444000  01452000  01455000  01461000  01469000  01472000  01474000  01483000  01484000  01493000


EXPCLASS
  0002 *00289000*

        03386000  03873000  03881000  03883000  03892000  03904000  03904200  03925000  03933000  03936300  03943300
        03990116  04015000  04074100  04112100  04202000  04381100


EXPLINK
  0002 *00153000*

        03909000  03915000 *04402000*


EXPLNK
  0121 *04027000*

       *04068000* 04402000
```

EXPNAME
  0115 *03848000*

       03863000 03894000

EXPONENT
  0088 *02728000*

       02741000 *02779000*

EXPONENTSIGN
  0088 *02728000*

       02734000 *02772000*

EXPR
  0006 *03620000*

       03655000 03872000 03976000 03990160 *04020000* 04192000 04198000 *04397000* 04404000 04544000 04550000
       04556000 04602000 04854000 04883550 04886100 04912400 04993500 05029000 05041000 05041800 05052000
       05300240 05335000 05359000 05521000

EXPRESLT
  0121 *04027000*

       *04074100**04083000* 04083000 04090000 *04112100* 04130000 *04130000* 04133000 04182000 *04183000**04201000*
       04201000 *04202000**04381100* 04400000

EXPRESULT
  0002 *00153000*

       03657000 03663000 *03866000* 03874000 *03878000**03883000**03887000**03892000* 03893000 03894050 03894060
       03894100 03894200 03897000 03900000 03902000 03903000 03904000 03908000 03912000 03913000 03914000
       03918000 03920000 03925000 03930000 03930600 03931000 03933000 03935000 03936200 03936300 03943200
       03943300 04194000 04198100 *04400000* 04602500 04887100 05521100

EXPT1
  0002 *00220000*

       04031000 *04209000* 04214000 04219000 04257000 04278000 04307000 04326000 04347000 04367000

EXPT2
  0002 *00220000*

       04031000 *04209000* 04214000 04219000 04223000 04257000 04278000 04294000 04295000 04296000 04297000
       04307000 04326000 04344000 04347000 04367000

EXPT3

0002 *00220000*

04031000 *04209000* 04231000 04232000 *04245500* 04247000 04252000 04253000 04273000 04274000 04302000
04303000 04323000 04324000 04342000 04344000 04363000 04365000

EXPV
0121 *04028000*

04183000 04293200 *04293200**04349000* 04351000 04354000 04356000 *04356000* 04358000 04401000

EXPVALUE
0002 *00153000*

03660000 03664100 *04401000* 04603500 04604030 05521200 05521400

EXTERNAL
0006 *05289000*

05300000 05557000

EXTID
0002 *00284000*

00861000 03613000 03616000 03894060 03894200 03908000 03914000 04114520 04114530 05297000

EXTRAINFO
0002 *00219000*

00715000 00822000 00823000 00835000 00867000 00895000 00897000 00907000 03386000 03679000 03688000
03815000 03832000 03872000 03880000 03897000 03900000 03904200 03913000 03918000 03922000 03927000
03937000 03940000 03958000 03990116 04014000 04438000 04465000 04479000 04509150 05258000 05457000
05529760

E1
0080 *02363000*

02365000 02373030 02384000 02390000 02410000

E2
0080 *02363000*

02367000 *02372010* 02386000 02396000 02402000 02416000

E3
0080 *02363000*

02406000 *02408500*

E4

0080 *02362950*

02364780 *02374100* 02374470 02374500

E4A
0080 *02362950*

02364320 02364384 02364400 02364480 02364600 02364640 02364700 *02374560*

E4B
0080 *02362902*

02374050 02414370 02414380

F
0006 *00647000*

00648000 00649000 00651000

F
0006 *00723140*

00723150 00723160

F
0006 *00706560*

00706570 00706600 00706700 00706730 00707000 00723100

F
0006 *01262000*

01262000 01264000 01359000 01364010 01365000 01404000 01437000 01520000 01599000 01624000 01625000
01758000 01768000 01778000

F
0036 *00944000*

00944000 01024000 01025000

F
0037 *01030000*

01032000 01054000 01055000 01069000 01126000

F
0034 *00915140*

00915240 00915260 00915300 00915320 00915460 00915480 *00915600* 00915600 00915720 *00915720* 00915760

```
      *00915760**00915780* 00915780 *00915800* 00915800  00916020 *00916020* 00916080 *00916080*

F
  0047 *01248000*
       01249000

F
  0046 *01237000*
       01241000  01253000

F
  0080 *02343000*
       02346000

F
  0080 *02362010*
       02362030  02362033  02362037  02362038  02362041  02362080  02362190

F
  0090 *03055000*
       03057000

F
  0090 *03037000*
       03039000

F
  0090 *03062000*
       03064000  03067000

F
  0092 *03070000*
       03070000

FALL
  0134 *04657000*
       04756000  04761500 *04763500*

FASTV
```

0002 *00208120*

02230130

FATAL
0006 *00615000*

00631000 00646200 00646600 01530000 05186000 05238100

FATALERR
0022 *00617000*

00622000

FAULT
0006 *04404050*

04404700 04415310 04415320

FD
0003 *00332000*

02972000

FETCH
0077 *02151000*

02157000 02167000 02168000 02169000 02185000 02191000 02196000 02198000 02201000 02205000 02206000
02216000 02217200 02217450 02219000 02220000 02221000 02222000 02224000 02229000 02230080 02230110
02230160 02230230 02234000

FF
0044 *01192000*

01232000

FF
0055 *01430000*

01434000

FFC
0090 *03033000*

03315000

FFF
0090 *03032000*

```
FID
  0090 *02986000*

       *03159000**03184054**03186000* 03203000 *03234000**03239000* 03243000  03244000  03247000 *03270000**03275000*
        03279000

FID
  0090 *03041000*

        03049000

FIELD
  0006 *00400010*

        00600060  00600090  04656500 *05603110**05603120* 05617000  05618000  05618100  05619000  05619010  05619020
        05619030  05619040  05619045

FIELD
  0134 *04656500*

        04737000 *04760900* 04762000 *04763000*

FIL
  0080 *02362010*

        02362010  02362036  02362037  02362040

FILEARRAYPRT
  0002 *00213000*

        03257000  03259000  04607000 *04607000* 04608000  04615000  04616000

FILECHECK
  0006 *04580000*

       *04587000**04591000* 04592000  04604050  04841000  04845000  04847000

FILECONTROL
  0006 *04619010*

        04619090  05554000  05570000  05573100  05573200  05573300

FILEF
  0093 *03098000*

        03204000
```

FILEID
  0002 *00287000*

        00749000 02187000 02188100 03144000 04588000 04604040

FILEINFO
  0002 *00208000*

        02195000 02197000 02200000 02230300 02238000 03159000 03160000 03161000 03162005 03162007 03164000

FILENAME
  0006 *04581000*

        04585000 04588000

FILEOPTION
  0006 *02058000*

        02082000 *02118000* 02244000

FILES
  0090 *02982000*

        03151000 03260000

FILETOG
  0002 *00168600*

        *01826000**01960000**02184000**02242000* 02854000 02862300 *04965000**04991000**05006000**05529140**05529210*
        *05529215**05529820*

FILETYPF
  0006 *04581000*

        04589000

FILNUM
  0090 *03041000*

        03046000

FILTYP
  0090 *03041000*

        03047000

FILTYP

0090 *02986000*

03156000 03161000 03177000 *03177000**03184020**03184040* 03196000 03203000

FIRST
0151 *05232050*

05235500 05270100 05270200 *05270200*

FIRSTCALL
0002 *00169071*

01018130 02034000 02052000 02286700 02380220 02389300 02401220 02409300 02432000 03385000 05614000
05615000

FIRSTSS
0114 *03793000*

03799000 03832000 03836000

FIXB
0006 *00408100*

00925030 00925040 01733100 *01993000* 02017000 04415270 04542000 04554000 04560000 04773130 04816137
04927000 04946600 04996000 05059000 05060000 05368000 05412000 05587000

FIXPARAMS
0006 *00796000*

00915000 00974000 00995000

FL
0134 *04655100*

04708520 04730137 04735750

FLAG
0006 *00601000*

00621000 00683000 00689100 00690000 00702000 00705100 00754000 00759000 00761000 00773200 00839000
00851000 00859000 00861100 00862000 00871000 00880000 00899000 00949200 00962000 01058000 01102000
01130500 01138000 01163100 01164000 01165100 01166000 01169000 01445000 01517000 01587000 01588300
01590000 01621000 01629000 01636000 01639000 01644000 01657000 01662000 01679000 01696000 01753000
01789000 01791100 01837000 01843000 01863000 01868000 01888000 01896000 01897000 01899000 01914000
01920000 01928000 01929000 01933000 01937000 01940000 01945000 01953000 01957000 01981000 01989000
02083100 02179000 02181000 02182000 02185000 02186000 02194000 02199000 02217200 02229000 02230020
02230045 02230120 02236000 02236010 02236020 02237021 02237050 02333050 02364320 02364384 02364400
02364480 02364600 02364640 02364700 02654000 02704000 02827200 02965000 03150000 03227000 03233000
03263000 03264000 03265000 03655000 03655500 03664100 03803000 03804000 03810000 03810010 03822000
03829000 03831000 03835000 03867000 03894200 03902000 03905000 03907000 03910000 03913000 03916000

03929000 03930800 03935000 03942000 03965000 03978000 03985000 03990165 03990195 04086000 04177600
04196000 04198100 04215000 04220000 04223000 04231000 04239000 04242000 04243000 04244000 04245000
04252000 04273000 04295000 04301000 04302000 04323000 04342000 04404200 04404250 04404650 04415100
04415110 04415130 04415160 04425000 04433000 04445000 04445200 04453000 04468000 04484000 04498000
04500300 04500320 04500420 04500600 04538000 04544000 04550000 04556000 04570000 04586500 04602000
04603500 04607000 04677500 04679000 04708515 04708550 04710500 04725550 04730100 04730135 04735600
04735715 04735750 04737600 04744110 04746000 04757000 04757420 04758000 04760000 04760500 04760600
04760800 04762000 04762510 04763500 04771500 04772500 04773030 04773090 04773100 04816000 04842000
04854000 04869000 04871200 04883550 04912400 04932000 04936000 04957050 04957055 04964100 04979000
04993510 04997000 05024000 05027000 05032000 05033000 05034000 05038000 05041000 05041800 05051100
05052000 05067000 05067280 05067330 05067370 05076000 05079000 05079900 05082000 05086000 05094000
05107060 05128000 05130150 05151000 05191100 05212005 05213000 05221000 05222000 05233000 05256000
05262300 05266000 05270100 05279000 05291000 05292000 05359000 05432000 05433000 05440000 05446000
05521000 05529310 05529350 05529390 05529450 05529510 05529520 05532000 05535130 05535480 05591000
05597000 05612200

FLAGI
0080 *02333050*

02364320 02364384 02364400 02364480 02364600 02364640 02364700

FLAGI1?
0080 *02333050*

02364320 02364384 02364400 02364480 02364600 02364640 02364700

FLAGLAST
0006 *05603010*

05603060 05603100 05619050

FLAGROUTINE
0006 *00600010*

00676010 00705110 00710010 00722010 00803010 00914010 00948010 01021959 01073010 01124010 01706110
01733200 01825000 01959000 01996010 02016010 02176010 02243010 03643000 03786000 03797000 03840000
03854000 03966000 03971010 03989010 03990047 03990815 03995010 04018010 04061000 04403000 04406025
04427025 04431010 04484010 04491000 04578000 04583010 04591010 04600010 04618010 04812010 04957110
05011000 05062010 05066010 05098010 05103010 05124010 05535045 05535605

FLAGROUTINECOUNTER
0002 *00161000*

00600060 *00600060* 00600065 00600090 00600100 *00600100*

FLAGROUTINEFORMAT
0004 *00333010*

00600050 00600080

FLG

0152 *05300180*

05300220 05300240 05300260

FLG1?
0152 *05300180*

05300220 05300240 05300260

FLOG
0006 *00608000*

01971000 01971010 01974500 01975000 02155000 02169000 02170000 02185010 02192000 02206000 02216000
02230150 02230180 02241000 02380300 02401300 02414250 02506030 02559000 02593000 02655500 02660000
02707500 02711000 02739000 02775000 02778100 02788000 02818030 02925100 03587000 03617000 03648000
03669000 03673000 03814000 03820000 03949000 03982000 03990136 03990180 04078000 04186000 04204000
04395000 04398000 04459000 04498100 04500725 04718000 04718100 04721500 04722000 04725500 04733500
04747000 04755500 04766500 04767500 04795000 04810000 04813000 04858000 04883100 04883700 04887100
04904000 04908000 04909000 04915000 04916100 04917100 04931000 04969000 04973000 04975000 05017010
05028010 05030000 05030100 05068100 05073000 05106000 05111000 05117000 05120000 05141000 05144000
05172000 05176000 05179000 05237000 05243000 05250000 05262000 05282000 05296000 05300180 05300220
05300240 05300260 05300290 05313600 05314300 05315600 05318000 05321000 05326000 05330000 05332000
05335000 05357000 05360000 05374000 05377000 05379000 05382000 05384000 05434000 05435000 05443000
05444000 05448000 05451200 05460000 05467100 05499110 05520100 05529140 05529250 05529420 05529620
05529635 05535090 05535110 05535225 05535235 05535450 05535570 05542000 05578000

FM
0135 *04655400*

04772610

FMINUS
0032 *00798000*

00814000 00824000 *00824000* 00827000 00842000 *00842000* 00845000 00847000 *00854100* 00854100 00855000
00856200 00872000 *00872000* 00875000 00876000

FMINUS
0041 *01127000*

01144000 01148000 01149000 01155000 *01163000*

FMT
0006 *00422140*

00422190 00422200

FMT
0039 *01057000*

01065000

FNFW
0002 *00159000*

03792000 *05108000* 05108200 05123000 05130000 05132000 05225000 05227000 05228000 *05446000* 05447000
05534000

FNFW
0006 *04429000*

04432000 04434000 04435000 04455000 04476000

FNFW
0114 *03792000*

03798000 03838000

FNFXT
0002 *00150000*

01851200 01852000 01931000 01938000 *02622200* 02623000 *02623000**02623100* 02644000 *02719000* 02754000
02755000 02758000 02759000 02767000 *02767000* 02787000 02787500 *02791000* 02791000 *02792000* 02792000
02796000 02797000 02797150 02797200 02797300 *02797300**02804000**02843000**02862400* 02862500 *02862500*
02862500 *02865000* 02866000 02870000 *02871000* 02874000 02875000 *02912000**02919000**03802100* 03802100
03810010 03812000 03813000 04068000 04071000 *04177500**04177700* 04180000 04183000 04388000 04404570
04413000 04415120 04495000 04884000 05012000 05039000 *05041100* 05041200 *05041200* 05041400 05067260
05067300 05069000 05107100 05115000 05117150 05141500 05142000 05144100 05145000 05243200 05244000
05257000 05313000 05313200 05316300 05375000 05380000 05385000 05436000 05446000 05529310 05529320
05535100 05535230 05535250 05535460

FNNHANG
0002 *00241000*

01615000 01616000 01634000 01639000 01646000 01657000

FNNHOLD
0002 *00238000*

01643000 01648000 03105000

FNNINDEX
0002 *00239000*

01637000 01643000 *01644000* 01647000 01648000 01649000 *01649000* 03103000 03105000

FNNPRT
0002 *00239000*

01636000 *01636000* 01642000 *01642000**01644000* 01647000 *03104000* 03104000

```
FORMAL
  0002 *00251000*

       00764020  01138000  01144000  01148000  01149000  01154000  01289000  01336000  01448000  03597000  03715000
       03803000  03829000  04101000  04415180  04443000  04500300  04504000  05094000  05278000  05535130  05535480


FORMALPP
  0006 *05100000*

       05125000  05129000  05224000  05533000

FORMARY
  0139 *04785020*

      *04886050* 04954100  04956800

FORMATER
  0006 *04620000*

       04773000  05575000

FORMATID
  0002 *00280000*

       00758000  04678500  04679000  04679400  04871000  04871100  04873000

FORMER
  0139 *04784000*

       04849000 *04866000*

FOUND
  0090 *03096000*

       03223000  03226000 *03229000*

FOUND
  0118 *03990040*

       03990080 *03990096*

FOUND
  0120 *03995000*

       04001000  04004000 *04006000*
```

```
FOUND
  0147 *05127100*

       05130125  05130140 *05130150*


FOUND1
  0084 *02523000*

       02572000 *02574000*


FOUND2
  0084 *02523000*

       02592000 *02594000*


FPB
  0090 *02984000*

       03127000  03343000


FPBBASE
  0005 *00374000*

      *00375000* 00377000  00378000  00379000  00380000


FPBI
  0090 *02986000*

      *03131000**03146000* 03146000  03166000  03187450  03191000  03203000  03354000


FPBSZ
  0090 *02988000*

       02988910 *03162005* 03203000 *03334000* 03335000  03339000  03340000  03343000  03344000  03361000


FPBVERSF
  0002 *00208170*

       03339100


FPBVERSION
  0002 *00208160*

       03339100


FPF
  0090 *02985000*

      *03127000**03203000* 03203000  03334000
```

FPHASF
  0134 *04644000*

       04717000

FPLP
  0088 *02729000*

       02750000 02761000

FPLUS2
  0002 *00355120*

       00812400 01732000

FPS
  0090 *02985000*

       03127000 03334000

FR
  0078 *02249500*

       02249500 02253700

FRACTION
  0088 *02728000*

       02735000 *02748000*

FREEFTOG
  0002 *00169077*

       02270500 02274100 02380000 02380125 02401000 02401125 02414175 02414225 02491600 02492450 02492550
       02655000 02707000 04667500 04669500 04670100 04767500

FREEREAD
  0139 *04785000*

       *04850100**04883000**04883650* 04883650 04912100 04917100 04950000 04954000 04956000 04957055

FROM
  0006 *03633000*

       03649100 03663000 03665010 03673100 *03673200* 03719000 03745000 03770000 03771000 03785100

FROM

0072 *01995000*

*01997000**01998000* 02001000 02004000 02007000 02009000

FROM
0084 *02540000*

02540000 02541000

FROM
0084 *02531000*

02532000 02533000

FROM
0084 *02524000*

02524000 02525000 02527000

FROM
0090 *03067000*

03077000

FTC
0002 *00329000*

04503020 04910000

FTF
0002 *00330000*

04415280 04941000 04949000

FUNCTION
0006 *05126000*

05135000 05480020 05558000

FUNID
0002 *00282000*

00706280 00859000 00861000 01018170 03213000 03614000 03616000 03894050 03894100 03909000 03912000
03990104 03990134 03996000 04115000 04117000 05106100 05108100 05116000 05129000 05224000 05226000
05503000 05529660

FUNVAR
0002 *00155000*

00785000 05107000 *05107040* 05107050 *05107100* 05133000 05505000 05529700

FX1
0002 *00163000*

00691540 *00692030* 01071900 *01077000* 01086000 01990000 *05012000* 05016000 05017010 05017020 05017030
05017050 05017055 05019000 05023000 05024000 05025200 05029400 *05035000* 05054000 05056000 05061000
05063660 *05069000* 05069200 05069300 05072000 05074000 05078200 05079400 05080000 05080500 05085000
05093000 *05244000* 05249000 05250000 05252000 05262200 05262300 05265000 05266000 05268000 05270100
05270500 05276000 05279000 *05375000* 05393000 05399000

FX2
0002 *00163000*

00691540 *00692030* 01072000 01082000 01084000 01085500 01090000 01095000 01095500 01098000 01099000
01108000 *01108000* 01109000 01110000 01113000 01114000 *05029000* 05033000 05061000 *05380000* 05387000
05399000

FX3
0002 *00163000*

00691540 *00692030* 01072100 01082000 01095000 01102000 *05385000* 05387000 05393000

F1
0002 *00192000*

02644000 02650000 02652000 *02665000* 02665000 *02672000* 02672000 *02751000* 02752000 02754000 02759000
*02762000**02768000* 02783000 02784000 *02784000* 02786000 02787500 02869000 02870000 02875000

F2
0002 *00192000*

02754000 02755000 *02870000* 02871000

F2TOG
0002 *00169060*

00792000 01728000 01730000 03129000

G
0006 *00649000*

00682000

G
0034 *00915140*

00915220 00915260 00915280 00915320 *00915360* 00915380 *00915600* 00915600 00915720 *00915720* 00915760
00915760 00915780 *00915780* 00915800 *00915800* 00916020 *00916020**00916080* 00916080 *00916120* 00916140

GBC
0002 *00320000*

01547000

GBW
0002 *00320000*

01547000

GEQL
0002 *00320000*

04228000 05389000

GET
0006 *00632000*

00634000 00635000 00665000 00685000 00700000 00702000 00705100 00737000 00739000 00739200 00744000
00815000 00833000 00834000 00899000 00909000 00910000 00914000 00972000 00991000 01007100 01130000
01135000 01159000 01162000 01276000 01280000 01287000 01306000 01311000 01325000 01331000 01442000
01445000 01517000 01588200 01589000 01589010 01601000 01627000 01628000 01634000 01639000 01646000
01651000 01673000 01677000 01707000 01708000 01791000 01800000 01978000 01980000 01987100 01988000
01989000 01990000 02188000 02190000 02569000 02862500 03106125 03585000 03586000 03609000 03649300
03802100 03855000 03909000 03915000 03939000 03952000 03963000 03965000 03972000 03976000 03990050
03990105 03990128 03996000 03996050 04075000 04082000 04082100 04086000 04100000 04114050 04114520
04117000 04117100 04123000 04126100 04136000 04155000 04404570 04415120 04415150 04415200 04415220
04417100 04422000 04425000 04432000 04467000 04496000 04498000 04500300 04500320 04500420 04500600
04503100 04503190 04535500 04537000 04591000 04679000 04773080 04773150 04871100 04874000 04875000
04879000 04880000 04956100 04956200 04956550 04965100 04965200 04983000 04993000 05016000 05017010
05017020 05017030 05017055 05023000 05024000 05025200 05029400 05069200 05069300 05078200 05107050
05107100 05108200 05117150 05133000 05144100 05183000 05196000 05223000 05243200 05249000 05250000
05252000 05262200 05262300 05265000 05266000 05270100 05279000 05297300 05313200 05436000 05439000
05446000 05503000 05529660 05529680 05529720 05529770 05535100 05535230 05535240 05535250 05535460

GETALL
0006 *00640000*

00748000 01082000 01095000 03143000 03596000 03646000 03798000 03984000 04442000 04455000 04884000
04990000 05074000 05130000 05225000 05447000

GETC
0006 *00646300*

00646400 00661000 00664000 00665000 00674000 00680000 00685000 00688000 00691900 00691950 00692000
00741000 00743000 00744000 01074000 01075000 01077000 01078000 01079000 01081000 01082000 01089000
01094000 01095000 01096000 01121000 01122000 01122500 05063400 05063630 05063650 05063660 05089000
05090000 05091000 05191000 05196000 05272000 05273000 05274000

GETCHAR
  0134 *04657500*

        04658000 04658500 04681000 04708520 04730120

GETCOL1
  0080 *02338000*

        02341000 02371002 02372040 02381000 02403000 02414260 02414382 02419100

GETFPB
  0005 *00365000*

        00367000 00370000 00375000

GETID
  0078 *02262700*

        02263300 02267500 02268400 02275160 02288500

GETSPACE
  0006 *00402000*

        00763000 *01126000**01130000* 01173000 01276000 01306000 01442000 01987000 03618000 03645000 03802100
        04081000 04404570 04415140 04498000 04500400 04500500 05041200 05446000 05535100 05535460

GETVOID
  0078 *02249500*

        02253100 02253400 02254300 02263900 02264800 02279300 02280300

GFC
  0002 *00321000*

        01548000 02009000

GFW
  0002 *00321000*

        00983000 01391000 01413000 01548000 02009000 04415080 04922000 04995000

GHX
  0002 *00225000*

        00227000 01745000 01771000

GIT

```
  0006 *01252000*

       *01260000* 01261000  01603000  01683000  01683100  01695100  02009000  04773130


GITSZ
  0090 *03055000*

        03060000  03061000  03158000  03184054  03185000


GLOBALENTER
  0006 *01768000*

       *01772000* 01777000  02187000  03593000  03602000  03607000  04587000


GLOBALNAME
  0002 *00169079*

        04500700  04503125  04509050  04513000  04516100  04883300  04883650  04957100


GLOBALNEXT
  0002 *00363000*

        01822000  01829000  01830000  01856000  01896000  01914000  01926000  01927000  01933000  01935000  01936000
        01940000  01943000  01944000  01948000  01949000  03668000  03673000  03864000  03867000  03947000  03949000
        03990136  03990170  03990180  04064000  04065000  04073000  04077000  04079000  04090000  04133000  04174000
        04185000  04189000  04194000  04203000  04404200  04404250  04404650  04415100  04415110  04415130  04420000
        04493000  04493100  04494000  04497000  04502000  04524000  04530000  04533000  04547000  04555000  04561000
        04562000  04563000  04564000  04566000  04571000  04573000  04576000  04577000  05039000  05300220  05300260
        05300290  05300300


GLOBALNEXTINFO
  0002 *00223000*

        01760000  01770000  01772000 *01776000* 01776000  03140000 *03381000*


GLOBALSEARCH
  0006 *01743000*

       *01749000* 01750000  02186000  03106125  03592000  03600000  03607000  04114520  04585000  04956550  05107040


GLOBALSTACKHEAD
  0002 *00227000*

        01745000  01771000  01772000


GNC
  0005 *00372000*

        00373000  00377000  00378000  00379000  00380000
```

GOTFOF
  0140 *04789000*

       04791000 04794000 *04805000*

GOTERR
  0140 *04789000*

       04793000 *04808000*

GOTOS
  0006 *05301000*

       05352000 05559000

GPHASE
  0134 *04644500*

       04717500

GROUPPRT
  0002 *00164000*

       00678000 00681000 *00685000* 00689000 *00689100**00690000* 00695000 01076000

GRTR
  0002 *00321000*

       03990550 04227000 04550000 05043000 05339000

GSEG
  0090 *02986000*

       03130000 03353000

GT
  0048 *01255000*

       01258000 01259000 01260000

H
  0034 *00915140*

       00915220 00915240 00915280 00915300 *00915360* 00915420 *00915520* 00915520 *00915600* 00915600 00915720
      *00915720**00915760* 00915760 00915780 *00915780**00915800* 00915800 *00916020* 00916020 *00916080* 00916080

HEADER
0002 *00293000*

00741000 05186200 05238300

HERE
0036 *00946000*

00988000 00991000

HERE
0121 *04022010*

04115010 *04124100*

HF
0134 *04656500*

04681020 04681030 04681100 04683000 *04684500**04719000* 04767500 04768000

HI
0006 *01404000*

01408100 01418000 01419000 01420000 01421000 01425000 *01426000*

HOLDID
0002 *00181000*

01971000 01974000 05025000 05025100 05029200 05029400 05030100

HOLLERITH
0081 *02629000*

02678000 02883000

HOLTOG
0002 *00169094*

02269600 02285500 02380100 02380125 02395000 02401100 02401125 02414200 02414225 02415000 03139000
04681020 04681030

HPHASE
0134 *04645000*

04690000 04693200 04719000 04719500 04740600 04760940

HUNT
0118 *03990040*

03990060 03990082

HV
0006 *00706700*

01018330

I
0002 *00151000*

00362000 00709000 00725000 00789000 00790000 00915140 00918000 00948000 01028000 01071200 01197000
01406000 01600000 01626000 01655000 01779000 *02230045* 02230050 *02230120* 02557000 02730000 02988000
03634000 03956000 03958000 03971000 03990030 03993000 04431000 04487100 *04587000* 04589000 04655500
*04870000**04871000* 04871100 04874000 04875000 04876000 04879000 04880000 *04884000* 04893000 04900000
04901000 04902000 04962000 05010600 05063250 05063620 05127100 05349000 05349200 *05451000* 05451100
05452000 05456000

I
0006 *00414000*

I
0006 *00641000*

00643000 *00643000* 00644000 *00644000* 00645000 *00645000*

I
0006 *00649000*

00653000 00677000 00688000

I
0006 *00646100*

00646200 *00646200* 00646200

I
0006 *00636000*

00638000 *00638000*

I
0006 *00646300*

00646400 *00646400*

I

```
0006 *00632000*
       00634000 *00634000* 00634000

I
0006 *00796000*
       00804000

I
0006 *01704000*
       01707000  01708000

I
0006 *01612000*
       01616000  01617000

I
0024 *00653000*
       00662000  00664000 *00664000* 00664000  00665000

I
0028 *00709000*
       00713000  00718000

I
0037 *01028000*
      *01036000* 01037000  01044000  01046000

I
0030 *00725000*
      *00736000* 00737000 *00738000* 00739000 *00739100* 00739200  00740000  00741000  00743000  00744000 *00749000*
       00752000  00758000  00760000  00762000

I
0035 *00918000*
       00924000

I
0034 *00915140*
       00915560  00915580  00915600  00915680  00915700  00915720 *00915840* 00915840  00915900  00915920  00915940
```

```
      00915960 *00916100* 00916140  00916160  00916300  00916400

I
  0036 *00948000*

      00954000  00985000  00989000  00995000  00995500 *01002000* 01003000  01011000  01012000  01019150  01019160
      01019170  01021600  01021700

I
  0040 *01071200*

      01080000  01081000  01082000  01087000  01089000  01093000  01094000  01095000  01096000  01119000  01121000
      01122000  01122550  01122600

I
  0045 *01197000*

      01200000  01202000 *01203000* 01204000

I
  0052 *01406000*

      01415000  01417000

I
  0059 *01600000*
      *01602000* 01608000

I
  0069 *01779000*
      *01791000* 01791100  01792000  01795000  01799000  01801000

I
  0061 *01626000*
      *01631000**01632000* 01632000 *01639000*

I
  0062 *01655000*
       01670000  01671000  01692000  01693000  01697000

I
  0088 *02730000*
      *02741000* 02744000  02745000  02746000 *02769000**02774000* 02784000 *02787000* 02787000
```

```
I
  0084 *02557000*

       *02568000* 02569000  02571000  02571100  02572000  02575000 *02577000* 02577000  02591000  02591100  02592000
        02595000  02597000 *02597000* 02597000  02601000

I
  0092 *03069000*

        03075000  03077000  03078000 *03078000* 03082000

I
  0090 *02988000*

        03069000  03140000 *03140000**03142000* 03142000  03143000  03211000  03282000  03283000  03297000  03299000
        03303000  03309000  03310000 *03335000* 03336000  03340000  03343000  03344000 *03356000* 03363000  03363400
        03363500  03363600

I
  0118 *03990030*

       *03990070* 03990070  03990080 *03990094* 03990098 *03990104**03990108* 03990112  03990132 *03990132* 03990145
        03990170 *03990170* 03990190

I
  0113 *03634000*

        03652000  03666000  03677000  03685000  03687000  03689000

I
  0117 *03971000*


I
  0120 *03993000*

       *04000000* 04001000  04002000  04004000 *04004000* 04007000  04012000

I
  0132 *04487100*

        04509100

I
  0131 *04431000*

        04436000  04438000  04440000  04447000  04463000  04464000  04465000  04466000  04467000  04477000  04480000
```

```
I
  0134 *04655500*

        04684000  04685000 *04685000**04686000**04689500* 04693000  04695500 *04696500* 04696500 *04698000**04719000*
       *04738500* 04739500  04740500 *04748800* 04749000 *04749200* 04749200  04749400  04772615  04772620 *04772635*
        04772635

I
  0141 *04962000*

        04980000  04982000  04984000  04987000  04993500 *04993500* 04993510  04993550  04993560  04993590  05003000


I
  0147 *05127100*

       *05130120* 05130130  05130140 *05130140* 05130150

I
  0142 *05010600*

       *05019000*

I
  0143 *05063250*

       *05063300**05063400* 05063400  05063450

I
  0144 *05063620*

        05063640  05063650 *05063650* 05063660  05063670

IC
  0002 *00221000*

        00362000  00634000  00638000  00643000  00644000  00645000  00646200  00646400  00646700  00715000  00822000
        00823000  00835000  00867000  00895000  00897000  00907000  00975000  00995500  01021450  01087000  01089000
        01610000  01687000  01739000  01740000  01747000  01748000  01763000  01764000  01765000  01773000  01774000
        01775000  02432300  03363500  03363600  03611000  03615000  03679000  03688000  03815000  03832000  03872000
        03880000  03897000  03900000  03904200  03913000  03918000  03922000  03927000  03937000  03940000  03958000
        03990116  03990122  04008000  04014000  04438000  04465000  04479000  04509200  04589000  05063670  05088000
        05090000  05191000  05191100  05195000  05258000  05271000  05273000  05286000  05457000  05529680  05529700
        05529760  05529790

ID
  0002 *00301000*
```

02155000 02261200 02362900 02557000 02862200 03801000 03993000 04073000 04077000 04185000 04404250
04415110 04494000 04573000 04860000 04883100 04883150 04918000 04969000 05068100 05106000 05115000
05144000 05166000 05176000 05243000 05296000 05311000 05435000 05444000 05535090 05535225 05535450
05539000

ID
0006 *00706010*

00706010 00706020

ID
0078 *02245600*

02246200 02247100

ID
0078 *02261200*

02262700 *02263300* 02263300 02263600 02264500 02267200 02267500 *02267500* 02268100 02268400 *02268400*
02271400 02272000 02272500 02272550 02272600 02272900 02273200 02273500 02273800 02274100 02274400
02275000 02275120 02275160 *02275160* 02275180 02275300 02275600 02275900 02276200 02276800 02277100
02277400 02277700 02278000 02278600 02278900 02279800 02280700 02282800 02283100 02285200 02285500
02285800 02286100 02286400 02287300 02287600 02288500 *02288500*

ID
0071 *01964000*

01968000

ID
0080 *02362900*

02364382 02364383 02364420 02364440 02364500 02364560 02364580 02364605 02364620 02364660 02364680
02371004 02372050 02382300 02382400 02404300 02404400 02414290 02414300 02419250 02419300

ID
0080 *02333100*

02333240 02333280 02333440

ID
0084 *02557000*

02570000 02571000 *02582000* 02591000

ID
0120 *03993000*

03996000 04001000 04002000 04004000

```
IDFN
  0132 *04488100*
         04488100  04488300  04581000  04619010

IDFNT
  0006 *00706150*
         00706260  00706380  00706450

IDINFO
  0002 *00162000*
        *02862000* 02862400  02862800 *02939100* 05041100 *05041100*

IDNM
  0090 *03042000*
         03051000

IDNM
  0090 *02986000*
         02993000 *03157000* 03158000 *03178000**03182000**03183000**03184010* 03184010  03184020 *03184020* 03184054
         03185000  03186000  03203000  03204000 *03223000* 03224000 *03226000* 03244000 *03267000*

IDSZ
  0090 *03041000*
         03043000  03050000  03051000

IDV
  0002 *00321000*
         01452000  01469000  01483000  03990250  04327000  04993560

IFS
  0006 *05353000*
         05417000  05574000

IJ
  0034 *00915140*
        *00915440* 00915460 *00915560* 00915580  00915660  00915680  00915700

IL
```

```
  0034 *00915160*

        00915840  00915860  00916100

IMPLICIT
  0006 *05529100*

        05529830  05577100

INAM
  0002 *00241030*

        03990070  03990084

INCLUDE
  0002 *00174300*

        02371004  02372050  02382400  02404400  02414300  02419300 *03437100*

INDX
  0032 *00802000*

       *00870000* 00892000

INDX
  0113 *03637200*

       *03665200* 03665200

INDX
  0132 *04487100*

       *04503180**04509300* 04509300

INF
  0032 *00801000*

       *00833000* 00843000  00846000  00854100  00854200  00856000  00856100  00873000  00877000  00894000

INFA
  0002 *00222000*

        00650000  01072000  01626000  03143000  03144000  03147000  03207000  03209000  03211000  03213000  03217000
        03243000  03252000  03576000  03637000  03793000  03984000  03986000  03988000  04431000 *04503190* 04509350
        04884000  04885000  04891000  04897000 *04900000* 04900000  04963000 *05069300* 05069400  05071200  05071300
        05071400  05074000  05076000  05078500  05080500  05082000  05085000  05086000  05086100  05088000  05093000
        05094000  05130000  05131000  05132000  05225000  05226000  05227000 *05265000* 05267100  05268000  05269000
        05270500  05271000  05276000  05278000 *05436000* 05437000  05447000  05447500  05448500  05453000  05458000
        05458400
```

```
INFA
  0006 *00641000*

       *00643000*

INFA
  0023 *00650000*

        00653000

INFA
  0024 *00653000*

       *00665000* 00668000  00672000

INFA
  0040 *01072000*

        01082000  01084000  01085500  01090000  01095000  01095500  01098000  01099000  01108000  01109000  01110000
        01113000  01114000

INFA
  0061 *01626000*

       *01627000* 01631000

INFA
  0112 *03576000*

        03596000  03596100  03597000  03598000  03601000  03605000  03606000 *03609000* 03610000

INFA
  0114 *03793000*

        03798000  03822000  03823000  03829000  03830000  03834000  03838000

INFA
  0113 *03637000*

        03646000  03647000  03703000  03713000  03715000  03716000  03719000  03725000  03738000  03743000  03747000
        03772000  03774000

INFA
  0131 *04431000*

       *04432000* 04433000  04434000  04442000  04443000  04444000  04445100 *04446000* 04446000  04448000  04455000

INFA
```

```
  0141 *04963000*
        04990000 *04993000* 04997000  04998000

INFADDR
  0006 *05063600*
        05063630  05063640 *05063670*

INFB
  0002 *00222000*
        00653000  00726000  01072100  03143000  03148000  03157000  03177000  03178000  03182000  03209000  03223000
        03224000  03226000  03227000  03233000  03251000  03576000  03637000  03793000  04431000  04884000  04904000
       *05070000* 05071400  05130000  05130120  05130130  05130140  05130150  05225000  05447000  05447500

INFB
  0006 *00641000*
       *00644000*

INFB
  0024 *00653000*

INFB
  0030 *00726000*
        00748000  00750000  00767000

INFB
  0040 *01072100*
        01082000  01095000  01102000

INFB
  0112 *03576000*
        03596000

INFB
  0113 *03637000*
        03646000  03648000  03669000  03673000

INFB
  0114 *03793000*
```

```
          03798000  03821000

INFB
  0131 *04431000*

          04442000  04445000  04445200  04455000  04459000

INFC
  0002 *00222000*

          00650000  00726000  01072200  01655000 *02190000* 02207000  02217000  02217450  02219000  02229000  02237000
          02243000  03143000  03155000  03156000  03177000  03210000  03231000  03232000  03576000  03637000  03793000
          03847000  03984000  03985000  03987000  04406000  04431000 *04875000* 04876000  04884000  04892000  04893000
         *04893000* 04896000  04902000  04963000  05074000  05079400  05079900  05080000  05130000  05225000  05228000
          05447000  05450000  05451000  05454000  05455000  05458400  05458600

INFC
  0006 *00641000*

         *00645000*

INFC
  0024 *00653000*

         *00665000* 00666000  00669000  00670000

INFC
  0023 *00650000*

          00653000 *00700000* 00702100

INFC
  0030 *00726000*

          00748000  00772000  00773000  00773200

INFC
  0040 *01072200*

          01082000  01083000  01084000  01085000  01091000  01095000  01097000  01099000  01100000  01111000  01112000


INFC
  0062 *01655000*

        *01677000* 01680000

INFC
```

```
  0112 *03576000*
        03596000  03596100  03607100

INFC
  0114 *03793000*
        03798000 *03836000**03837000* 03837000  03838000

INFC
  0113 *03637000*
        03646000  03649000  03650000  03651000  03714000

INFC
  0115 *03847000*
       *03855000* 03856000  03858000  03859000  03859500 *03952000**03953000* 03953000  03955000

INFC
  0130 *04406000*
       *04422000* 04423000  04424000

INFC
  0131 *04431000*
        04442000  04447000  04448000  04455000  04456000  04458000  04460000  04462000  04473000 *04473000* 04474000
       *04474000* 04476000

INFC
  0141 *04963000*
        04990000

INFD
  0113 *03637300*
       *03649300* 03665300

INFO
  0002 *00217000*
        00634000  00638000  00643000  00644000  00645000  00975000  00995500  01610000  01687000  01739000  01740000
        01747000  01748000  01763000  01764000  01765000  01773000  01774000  01775000  03611000  03615000  03990122
        04008000  04589000  05529680  05529700  05529790

INFOF
```

```
  0031 *00728000*

       00767000


INFOT
  0031 *00732000*

       00735000


INITIALIZATION
  0006 *03368000*

       03572000  05611000


INITIALNCR
  0002 *00175000*

       02428000  02647000 *03375000*


INITIALSEGNO
  0002 *00149000*

      *00788000* 00956000  00966000


INLINFFILE
  0006 *04593000*

       04619000  04619060  04826000  04850000


INLINEINT
  0002 *00169310*

       03484100  03563910  03563940  03563950  03990050  03990070  03990084  03990098  03990132


INP
  0006 *00706730*

      *00706760* 00706820  01018330


INSERTCOP
  0002 *00241240*

       02364350  02364383  02364810  02374300  02374680


INSERTDEPTH
  0002 *00174200*

       00241240  00241260  00241280  00241300  00241320  02364160  02364300 *02364300* 02364340  02364350  02364380
```

02364381 02364383 02364385 02364460 02364520 02364605 02364660 02364680 02364720 02364740 02364780
02364810 02364860 02374150 *02374150* 02374300 02374450 02374470 02374500 *02374600* 02374600 02374680
02414370 02414380 *03437110*

INSERTF 0002 *00241280*

02364460 02364520 02364740 02374450

INSERTINFO
0002 *00241360*

02364340 02364350 02364380 02364381 02364383 02364385 02364460 02364520 02364605 02364660 02364680
02364740 02364780 02364810 02374300 02374450 02374470 02374500 02374680 02414370 02414380

INSERTINX
0002 *00241300*

02364340 02364780 02374470 02414370

INSERTMAX
0002 *00241220*

00241360 02364300

INSERTMID
0002 *00241260*

02364380 02364381 02364385 02364740 02374450

INSERTSEQ
0002 *00241320*

02364605 02364660 02364680 02374500 02414380

INT
0002 *00233000*

03223000 03226000 03230000 03234000 03245000 03247000 03248000 03267000 03268000 03270000 03279000
03485000 03563940 03990098 03990130 04001000 04004000 04007000 05130120 05130140 05130150

INTCLASS
0002 *00241010*

03990122 03990200 04008000

INTEGERS
0006 *05484000*

05560000

```
INTFID
  0115 *03849000*

       *03859500* 03873000  03903100  03930100  03930650  03936300  03964000  03964500

INTID
  0002 *00165000*

        01019170  03564000  05529220

INTINLINE
  0002 *00241010*

        03990130

INTLOC
  0090 *02989000*

        03234000  03239000  03270000  03275000

INTNUM
  0002 *00241020*

        03234000  03270000

INTPARMCLASS
  0002 *00241010*

        03990098  04011000

INTPARMS
  0002 *00241020*

        03230000  03990108  03990190  04007000

INTPRT
  0002 *00241020*

        03247000  03279000

INTRFUNID
  0002 *00283000*

        01594000  04055000  04364000  04415300  04834000  04943000  04954000  04954050  04954100  04954150  04954200
        04956700  04956750  04956800  04956925  04956930  05535160  05535520

INTSEEN
```

0002 *00241030*

03990130 04007000 05130150

INTX
0002 *00241030*

03563950 03990098

INTYPE
0002 *00295000*

00896000 01452000 01454000 01469000 01470000 01483000 01490000 01851300 01897000 02864000 03564000
03810000 03877000 03938000 03941000 03990160 04177400 04257000 04278000 04307000 04326000 04344000
04347000 04367000 04367500 04985000 04993530 04993560 05067240 05256000 05484000 05529290 05560000

INX
0002 *00321000*

01341000 01464000 01503000 01513000 03752000 03777000 04148000 04163000 04503040

INXFIL
0002 *00213000*

02180000 *02180000* 02188000 02195000 02197000 02200000 02230300 02238000

IOCOMMAND
0006 *04774000*

04958000 05544000 05550000 05564000 05566000 05567000 05573000

IOLIST
0006 *04486000*

04529000 04579000 04924000

IP
0002 *00235000*

04063000 04063000 04074100 04207000 04210000 04380000 *04380000**04383000* 04383000 04384000 04395000
04396000

IPHASE
0134 *04645500*

04721000

IR

0002 *00221000*

00362000 00634000 00638000 00643000 00644000 00645000 00646200 00646400 00646700 00715000 00822000
00823000 00835000 00867000 00895000 00897000 00907000 00975000 00995500 01021450 01087000 01089000
01610000 01687000 01739000 01740000 01747000 01748000 01763000 01764000 01765000 01773000 01774000
01775000 02432300 03363500 03363600 03611000 03615000 03679000 03688000 03815000 03832000 03872000
03880000 03897000 03900000 03904200 03913000 03918000 03922000 03927000 03937000 03940000 03958000
03990116 03990122 04008000 04014000 04438000 04465000 04479000 04509150 04589000 05063670 05088000
05090000 05191000 05191100 05195000 05258000 05271000 05273000 05286000 05457000 05529680 05529700
05529760 05529790

ISn
0002 *00322000*

01454000 01470000 01490000 04404570

ISN
0002 *00322000*

03623000 04606000 05336000

ISO
0002 *00328000*

03625000

IT
0002 *00235000*

00706750 *00706750**00706760* 00706780 *00723630* 00723630 *00723650* 00723680 *00916220* 00916220 *00916260*
00916320 *01018170* 01018200 *01018260* 03644000 *03654000* 03654000 03664100 03665000 *03785000* 03853000
03853000 03893000 *03945000* 03945000 03948000 03956000 *03967000* 03990200 04058000 *04058000* 04062000
04062000 04075000 *04075000* 04123000 04181000 *04181000**04192000* 04192000 04196000 04200000 04208100
04209000 *04216000* 04216000 04221000 *04221000* 04250000 *04250000**04252000* 04252000 *04257000* 04257000
04258000 *04258000**04260020* 04260020 *04265000* 04265000 04267000 *04267000**04269000* 04269000 *04273000*
04273000 *04278000* 04278000 04279000 *04279000* 04286000 *04286000* 04288000 *04288000**04290000* 04290000
04291015 04291015 *04302000* 04302000 *04307000* 04307000 04308000 *04308000**04315000* 04315000 *04317000*
04317000 04323000 *04323000* 04327000 *04327000* 04328000 *04328000**04334000* 04334000 *04336000* 04336000
04342000 04342000 04347000 *04347000* 04348000 *04348000* 04367000 *04367000**04368000* 04368000 04369000
*04369000**04370000* 04370000 04371000 *04371000**04372000* 04372000 *04373000* 04373000 *04374000* 04374000
04375000 *04375000* 04397000 04398000 *04399000**05334000**05351000*

IU
0034 *00915160*

00915840 00915860 00916100

IX
0006 *01437000*

01442000 01445000 01453000 01458000 01517000

```
J
  0002 *00151000*

       *00815000* 00819000  00820000  00905000  00915140  01600000  01613000  01655000  01706000  01752000  01759000
        01769000  02990000  04655000  04773020 *05349200* 05349300 *05436000* 05438000 *05450000* 05456000

J
  0034 *00915140*

       *00915360* 00915560  00915620  00915660  00915760  00915840  00915860 *00915860* 00915900  00915920  00915940
       *00916100**00916160* 00916300

J
  0059 *01600000*

       *01603000* 01607000

J
  0068 *01769000*

       *01771000* 01772000 *01772000* 01773000  01774000 *01774000* 01774000  01775000 *01775000* 01775000

J
  0063 *01706000*

       *01718000* 01720000  01727000

J
  0066 *01752000*

        01754000

J
  0067 *01759000*

       *01761000**01762000* 01762000  01763000  01764000 *01764000* 01764000  01765000 *01765000*

J
  0062 *01655000*

        01657000 *01657000**01683000* 01685000

J
  0060 *01613000*

        01614000  01615000  01616000  01617000
```

```
J
  0092 *03069000*

        03075000  03077000


J
  0090 *02990000*

        03069000 *03155000* 03159000  03160000  03161000  03162005 *03162007* 03162007  03164000 *03164000* 03164050
        03165000  03169000  03170000  03171000  03172000  03173000 *03184050**03259000*


J
  0134 *04655000*

       *04689500* 04692000  04696000 *04696000* 04739500 *04739500* 04740100  04740110 *04748800* 04748800  04749200
        04749400  04749800  04772620 *04772620* 04772625  04772630  04772635


J
  0138 *04773020*

        04773130 *04773130*


JPHASE
  0134 *04646000*

        04729500


JUNK
  0002 *00290055*

        01459000  01463000  01495000  01497000  01499000  01502000  01508000  01512000  03623000  03628000  04124000
        05336000  05338000  05341000  05506000


K
  0006 *00706525*

        00706525  00706530


K
  0006 *01778000*

        01780000  01788000 *01788000* 01788000  01790000  01811000  01812100


K
  0034 *00915140*

       *00915360* 00915420  00915440 *00915520* 00915520 *00915560* 00915800 *00915800* 00915820  00915840  00915860
       *00915960**00916000* 00916000  00916020  00916040  00916080
```

```
KEEP
  0077 *02140000*

       *02207000**02217000* 02217050  02217200 *02217250* 02217250  02217275 *02217275* 02217350 *02217350* 02217450
        02219000  02220000  02229000  02230020  02237040

KK
  0134 *04655100*

       *04730120* 04735610

KLASS
  0002 *00234000*

        00723670  00769000  00916300  01018120  01018360  03394000

KOM
  0002 *00319000*

        03990450  04411000  04412000  05214000  05300270  05477000  05491000  05492000  05513000  05535580

L
  0007 *00426000*

        00430000

L
  0006 *00394500*


L
  0006 *00412000*


L
  0006 *01365000*
        01367000  01368000

L
  0006 *01261110*
        01261120

L
  0006 *01177000*
```

01181000 01183100

L
0006 *03991000*

03996000 *03996000* 03996050 04008000 04009000

L
0023 *00650100*

00653000

L
0024 *00653000*

00665000 *00665000* 00670000 00672000

L
0029 *00723470*

00723610 00723680

L
0034 *00915140*

00915380 00915420 *00915560* 00915780 *00915780* 00915820 00915840 00915860 *00916160* 00916320

L
0045 *01198000*

01202000 *01204000*

L
0078 *02249800*

02250400

L
0078 *02254900*

02255500 02255800

L
0084 *02523200*

02523300 02523400

L

0081 *02802000*

L
0082 *02489400*
02490000 02496000 02497000

L
0136 *04663500*
04664500

L
0145 *05066000*
05067280 05067340 *05067390*

L
0157 *05529120*
05529310 05529360 *05529400*

LA
0002 *00228000*
02245900 02254900 02997000 04924000 *04924000*

LA
0078 *02254900*
02257000

LA
0078 *02245900*
02248300

LA
0090 *02997000*
03228000 *03254000*

LAB
0076 *02063000*
02067000 02072000

LAB
  0071 *01964000*

       01966000

LABELBRANCH
  0006 *01778000*

       01814000 05307000 05317000 05349200 05390000 05391000 05396000 05397000 05402000 05403000 05410000
       05411000 05414000

LABELID
  0002 *00281000*

       00706220 00752000 01588100 01588200 01588400 01673000 01791100 01811000 01812100 01976000 01977100
       03582000 03866000 04465000 04480000 04773060 04773080 04773175 04800100 05141500 05391200 05397200
       05403200 05521200 05535560

LABELMOM
  0002 *00160000*

      00785000 00809000 00829000 04453000 *04453000* 05520100 05523000

LABELR
  0006 *02059000*

      02080000 02117000 02808000 02835000

LABF
  0031 *00733000*

       00755000

LABL
  0002 *00174000*

       01780000 *02079000* 02081000 02082000 02083100 02089000 02092000 04677500 04678000 04678500 04679400
      04679500 04773040 04773050 04773070 04773175 *04816100**04964200**05067100**05128100**05151100**05213100*
      *05222100**05233100**05292100* 05365100 *05365100**05366100**05433100**05532100* 05580000 05584000 05591000
      05592100

LABL
  0069 *01780000*

       01788000 01790000 01811000 01812100

LABPRT

0062 *01655000*

*01677000**01679000* 01684000

LABX

0032 *00798000*

00806000 00810000 *00828000* 00828000

LADDR

0002 *00229000*

04415170 04415210 *04503010**04503055* 04507000

LADR1

0002 *00229000*

04415030 04415270 *04920000* 04927000 *05041300**05041600* 05059000 *05442000**05445000* 05445000 05462000

LADR2

0002 *00229000*

04487000 *05041400**05041700* 05058000 *05436000* 05437000 05439000 05463000

LADR2

0132 *04487000*

04527000 04531000 04542000 04546000 04551000

LADR3

0002 *00229000*

04541000 04554000 *05045000* 05060000

LADR4

0002 *00229000*

04549000 04559000

LADR5

0002 *00229000*

04553000 04560000

LAST

0023 *00650100*

00653000

```
LAST
  0024 *00653000*

       *00661000* 00662000

LAST
  0040 *01071500*

       01078000  01080000  01093000  01119000  01122550  01122600

LAST
  0144 *05063620*

       *05063630* 05063640

LASTADDR
  0002 *00242700*

       02432200 *02432200*

LASTC
  0002 *00244000*

       00661000  01078000  01122000  05063630  05063670  05088000  05191000  05195000  05271000  05286000

LASTERR
  0002 *00169210*

       00598180  00599500  03374200  05619050

LASTF
  0030 *00725100*

       00764150 *00764200* 00764260

LASTLINF
  0002 *00169097*

       00723640  00723690  00723720  00916240  00916340  00916400  01018280

LASTMODF
  0002 *00166100*

       02265700 *02272510**02272560**03387100*

LASTNFXT
```

0002 *00176502*

*01019190**03381200* 05529130 05529131 *05578100*

LASTSEQ
0002 *00169210*

00598040 00794400 02380200 02380300 02401200 02401300 02420011 02420020 02420040 02420050 02420120
02432300 03374100

LAX
0002 *00212000*

00925030 *01531000* 01532000 01708400 04415030 04527000 04541000 04553000 04816150 04920000 04946500
04976000 05041300 05041600 05045000 05048000 05364000 05412000

LB
0024 *00655000*

00666000 00670000

LBLSHFT
0006 *01174000*

01178000 01184000 01185000 01586000 01788000 01975000 04868000

LBRANCH
0002 *00211000*

00212000 00789000 01530000 01692000

LC
0078 *02254900*

02257600 02258500 02259100

LC
0078 *02249800*

02251300 02252900

LD
0078 *02249800*

02252500 02253400

LE
0078 *02249800*

02251900 02253100

LE
0078 *02245900*

02247200 02247800 02247900

LENGTH
0002 *00164000*

00656000 00657000 *00678000**00693000* 00693000 00693500 *00693500* 00694150 *00694150* 00694200 00696100
00701000 00702100 *00702100* 00703000 *00703000* 00705100 01116000 *01116000**03812000* 03813000 *03813000*
*03831000**03832000**03834500* 03834500 03835000 03836000 *05079200* 05079400 *05245000**05257000* 05257000
05267100 05267100 05268000 05277000

LEQL
0002 *00322000*

04225000 05401000

LESS
0002 *00323000*

03990310 03990550 04224000 05337000 05409000

LEVEL
0006 *04486000*

04498100 *04498100* 04528000 *04528000* 04529000

LF
0006 *00392600*

02364720 02364740 02364780 02374100 02374450 02374470 02414370

LIBADD
0080 *02362950*

02364100 02414300 02419300

LIBRARYFIL
0006 *00392400*

00392600 02364720 02364740 02364780 02374100 02374450 02374470 02414370

LIBTAPE
0002 *00169053*

02270200 02275100 02364860

LIMIT
0002 *00161010*

00606500 00613500 *02281000* 02281300 02281600 *03385360*

LINDX
0002 *00229000*

04415120 *04415120* 04415140 *04415140* 04415150 04415200 04415220 *04495000* 04496000 04498000 04500300
04500320 04500400 *04500400* 04500420 *04500500* 04500500 04500600 04503100 04513100 04516000 04535500
04537000 04545000 04557000 04558000

LINE
0006 *00387000*

00391000 00422110 00422190 00422200 00723520 00723640 00723720 00916240 00916400 00916460 01018130
01018140 01018150 01018170 01018200 01018210 01018375 01019000 02029000 02031000 02032000 02278600
02287300 02287600 04772610 04772620 04772645 05617000 05618100 05619000 05619020 05619040 05619050

LINEDICT
0002 *00242100*

01021450 03363500 03363600 03363700

LINESFG
0002 *00242200*

00794400 01021700 02432300

LINESIZF
0006 *00386000*

00387000 00388000

LINFA
0002 *00229000*

04415150 04415165 04415170 04415180 04415230 *04498000**04500600* 04500655 04503010 04503055 04504000
04517000 04535500 04536000

LINK
0002 *00254000*

00664000 00665000 00744000 00805000 00935000 00989000 00995500 01081000 01082000 01094000 01095000
01121000 01538000 01689000 01740000 01748000 *01761000**01771000* 02001000 *02207000**02217000**02217450*
*02219000**02229000* 02237000 03156000 03177000 03990128 04030000 04404150 04406000 04589000 05063650
05063660 05191000

```
LINK
  0006 *03633000*

       *03645000* 03645000  03646000  03781000  03782000

LINK
  0006 *03969000*

        03972000  03984000

LINK
  0006 *03788000*

        03792000  03798000  03799500 *03799500* 03799500  03838000

LINK
  0006 *03842000*

        03855000  03939000  03952000  03955000  03963000  03965000

LINK
  0006 *04959000*

        04965100  04965200  04990000  04993000  04998000  05000000
```

```
LINK
  0129 *04404150*

        04404570 *04404570*

LINK
  0121 *04030000*

       *04068000**04071000* 04075000  04081000 *04081000* 04082000  04082100  04085000  04086000  04092000  04094000
        04100000  04114050  04114100  04114520 *04114530* 04114530  04115010  04116000 *04116000* 04117000 *04117000*
        04117100  04120000  04122000  04123000  04126100  04128000  04136000  04155000  04169000  04349000 *04388000*

LINK
  0130 *04406000*

       *04417000* 04417100  04421000  04422000  04423000  04425000  04426000

LINKC
  0090 *03031000*

        03317000
```

LINKF
  0090 *03030000*

LINKLIST
  0002 *00169210*

       00599500 00706260 00706380 02420060 02426030

LISTART
  0002 *00232000*

       04415090 04415240 04519000 *04923000*

LISTEL
  0134 *04656500*

       *04735740**04762500* 04772000

LISTER
  0139 *04784000*

       04869000 04871200 04882000 04883000 04883700 *04912000*

LISTER1
  0139 *04784200*

       04888000 *04912700*

LISTID
  0002 *00199300*

       *03567100**04493200* 04493200 *04883350* 04883350

LISTLIBTOG
  0002 *00169063*

       02277700

LISTOG
  0002 *00169062*

       00422275 00598120 00604000 00611000 00781000 01005000 01018370 01019000 01050000 02083000 02086500
       02088000 02093000 02095000 02177000 02262775 02269900 02272300 02274700 02275300 02276500 02289450
       02364890 02380210 02401210 02420100 02432000 02493200 02506010 02662000 02713000 02818010 03384000
       03385100 04671500 04769500

LISTPTOG
  0002 *00169095*

        02270500 02286400 02389200 02409200

LISTSID
  0002 *00290050*

        04493200 04883350

LND
  0002 *00322000*

        03630000 03990270 04219000 04245500 04404425 04404625

LNG
  0002 *00322000*

        00925020 01708300 03990290 03990555 04223000 04816140

LNK
  0006 *03990010*

        03990050 03990088 03990100 03990104 *03990104* 03990105 03990108 03990122 03990127 *03990130* 03990150
        03990210 03990490

LNK
  0063 *01706000*

        01708000 01711000 01713000 01718000 *01719000* 01727000

LNK
  0090 *02988000*

        03314000 03316000 03317000

LO
  0006 *01404000*

        01408100 01426000

LOAD
  0032 *00802000*

        00852000 00859000 00861000 00862000 00880000

LOCALNAME

0002 *00169081*

04492100 04493300 04500700 04503125 04509050 04513000 04516100

LOCALS
0002 *00161000*

00396500 00396510 *00690000* 00690000 *00690000* 00691510 *00692010**00791000* 00924000 00954000 01071000
01079000 01089000 *01099000**01100000* 01101000 *01138000* 01138000 *01138000* 01164000 *01164000* 01164000
01164000 01166000 *01166000* 03831000 *03831000* 03831000 *03831000* 04453000 *04453000* 04979000 *04979000*
05079000 *05079000* 05079200

LOD
0002 *00323000*

00764260 00845000 00855000 01282000 01313000 01342000 01344000 01463000 01502000 01512000 01606000
03627000 03726000 03739000 03753000 03755000 03778000 03780000 03888000 04105000 04138000 04151000
04164000 04166000 04415170 04415280 04503010 04503050 04503055 04617000 04880000 04897000 04939000
04941000 04949000 04956200 04956550 05535150 05535500 05535580

LOGICAL
0006 *05482000*

05561000

LOGIFADR
0154 *05354000*

05364000 05368000

LOGIFTOG
0002 *00169074*

02827100 04816000 05032000 05067000 05213000 05233000 05292000 05365000 05367000 05433000

LOGTYPE
0002 *00298000*

01084000 01085500 01135000 01443000 01450000 01471000 01481000 01913400 02920000 03834000 04214000
04219000 04223000 04250000 04295000 04993000 04993500 05078000 05267100 05361000 05482000 05506000
05529230 05561000

LOOKFORINTRINSIC
0006 *00394500*

03990088 *03991000**03996000* 04019000 04116000

LOOP
0077 *02159000*

```
          02161000  02162000

LOOP
  0076 *02062000*

       *02078000* 02084000  02086500

LOOP
  0082 *02487400*

       *02491400* 02493400

LOOP
  0081 *02450000*

        02458000  02459000  02496000  02497000

LOOP
  0083 *02498000*

       *02501000* 02506040

LOOP
  0081 *02803000*

       *02813000* 02818040

LOOP
  0090 *03090000*

        03092000  03093000

LOOP
  0115 *03846000*

       *03861000* 03947000

LOOP
  0118 *03990040*

       *03990140* 03990170

LOOP
  0114 *03794000*

       *03800000* 03819000

LOOP
```

```
  0121 *04021000*
        04064000 *04070000* 04205000  04389000  04394000

LOOP
  0136 *04660000*
       *04667000* 04672000

LOOP
  0132 *04488000*
        04522000  04534000 *04562000*

LOOP
  0146 *05103000*
       *05113000* 05119000

LOOP
  0145 *05066000*
       *05068000* 05098000

LOOP
  0149 *05163000*
       *05171000* 05197000

LOOPO
  0083 *02498000*
       *02498100* 02506030

LOOPO
  0081 *02801000*
       *02810100* 02818030

LOR
  0002 *00323000*
        03990430  04214000  04245500  05340000

LOS
  0002 *00361000*
        00376000  00378000  00387000
```

```
LOWERBOUND
  0002 *00164000*
         00655000  00666000  00670000 *00679000* 00693000 *01117000* 01117000

LP
  0078 *02261800*
        *02271400* 02275200  02288800

LP
  0078 *02245900*
         02246500  02246800

LP
  0080 *02333180*
         02333220

LP
  0134 *04657000*
        *04725000* 04754000

LPAREN
  0002 *00302000*
         01914000  02930000  03990136  04079000  04189000  04404200  04415100  04420000  04502000  04524000  04840000
         04842000  04883500  04912300  04918000  05014000  05071300  05072000  05110000  05237000  05247000  05314300
         05321000  05357000  05529420

LPGLOBAL
  0002 *00195000*
         02575000  03410000

LPHASF
  0134 *04646500*
         04730500

LPPHRASF
  0134 *04642520*
         04726530  04740100
```

LQ
  0080 *02333180*

        02333360 02333370 02333400

LQV
  0002 *00323000*

        03990390

LST
  0070 *01817000*

        01839000 01865000 *01878000* 01883100 01890000 *01892000* 01922000 01955000

LSTA
  0002 *00206000*

        00691520 *00692020* 01071400 *01083000**01084000* 01088000 *01097000* 01116000 03112000 *03112000* 04932000
       *04932000**04935000* 04935000 *04936000* 04939000

LSTI
  0002 *00206000*

        00691530 *00692030* 01071800 *01076000* 01108000 03109000 03111000 04933000 04934000 *04936000* 04938000
        04939000 *04940000* 04940000

LSTMAX
  0002 *00204000*

        00205000 01836000 01842000 01862000 01868000 01888000 01919000 01929000 01953000 01957000 04500725
        04685500 04697500 04726540 04766000 04933000 05448000 05451100

LSTP
  0002 *00205000*

        03111000 04934000 04938000

LSTS
  0002 *00206000*

        00691520 *00692010* 01071250 *01082000* 01085000 *01095000* 01112000 01114000 *01122000* 01122500 01122600
       01828000 01836000 *01838000* 01838000 *01839000* 01839000 01842000 01844000 *01844000* 01845000 *01845000*
       01846100 01846100 01862000 *01864000* 01864000 *01865000* 01865000 01868000 01869000 *01869000**01870000*
        01870000 *01871000* 01871000 01888000 01889000 *01889000**01890000* 01890000 01919000 01921000 *01921000*
       01922000 01922000 01929000 01930000 *01930000**01931000* 01931000 01938000 *01938000**01947000**01953000*
        01953000 *01954000* 01954000 01955000 *01955000**01957000* 01957000 01958000 *01958000* 04933000 04938000
       04938000 04940000 *05442000* 05448000 *05448000* 05448500 05451100 05452000 *05452000**05457000* 05457000
        05458000 *05458000* 05458400 05461000 05463000

LSTT
0002 *00205000*

01838000 01839000 01844000 01845000 01864000 01865000 01869000 01870000 01871000 01889000 01890000
01921000 01922000 01930000 01931000 01938000 01954000 01955000 01958000 03106205 03106210 03116000
03124000 03640000 03664100 03665000 03689000 03852000 03893000 03958000 03990045 03990200 04026000
04058000 04062000 04075000 04123000 04181000 04192000 04196000 04200000 04209000 04250000 04258000
04260020 04279000 04291015 04308000 04328000 04348000 04367000 04368000 04369000 04372000 04397000
04398000 04438000 04440000 04464000 04466000 04467000 04480000 04654000 04680500 04684000 04686000
04687560 04687590 04687600 04690000 04692000 04695500 04698000 04719500 04726530 04732000 04739500
04740100 04740110 04747400 04748800 04749400 04749800 04758500 04772000 04772600 04772620 04938000
04961000 04970000 04982000 04984000 04987000 05115000 05116000 05327000 05349200 05442000 05448500
05452000 05457000 05458000 05458400 05461000 05462000 05463000 05529740

LTYP
0070 *01817000*

01828000 01834000 01838000 *01840000* 01860000 01864000 *01866000* 01876000 *01879000* 01883000 01889000
01892000 01917000 01921000 *01923000* 01951000 01954000

L1
0006 *00422140*

00422160 *00422170* 00422180 *00422180* 00422190 00422200

L1
0006 *00706015*

00706080 00706085

L1
0034 *00915200*

00915560 00915900 00915920

L1
0080 *02362020*

02362037 02362040 02362080 02362190

L1
0081 *02802100*

02802300 *02863000*

L1
0158 *05535040*

05535080 05535180

```
L10
  0081 *02802100*
         02802300 *02943100*

L11
  0034 *00915200*
        *00915420* 00915520

L11
  0081 *02802100*
         02802300 *02945100*

L12
  0034 *00915200*
        *00915580* 00915600

L12
  0081 *02802100*
         02802300 *02946100*

L13
  0034 *00915200*
        *00915700* 00915720

L13
  0081 *02802100*
         02802300 *02947100*

L14
  0034 *00915200*
        *00915660* 00915760

L14
  0081 *02802100*
         02802300 *02948100*

L15
  0081 *02802100*
         02802300 *02949100*
```

```
L16
  0081 *02802100*
         02802400 *02951100*

L17
  0081 *02802100*
         02802400 *02960100*

L18
  0081 *02802200*
         02802400 *02962100*

L19
  0081 *02802200*
         02802400 *02963100*

L2
  0006 *00706015*
         00706085
```

```
L2
  0006 *00422140*
         00422180

L2
  0034 *00915200*
         00915780 *00915780* 00915820

L2
  0080 *02362020*
         02362060  02362210

L2
  0081 *02802100*
         02802300 *02898100*

L2
```

0158 *05535040*

05535440 05535595

L20
0081 *02802200*

02802400 *02964100*

L21
0081 *02802200*

02802400 *02965100*

L3
0006 *00422140*

00422180

L3
0034 *00915200*

00915800 *00915800*

L3
0080 *02362020*

02362120

L3
0081 *02802100*

02802300 *02899100*

L3
0158 *05535040*

05535540 05535585

L4
0006 *00422140*

00422180

L4
0034 *00915200*

00915540 *00915900* 00916100

```
L4
  0080 *02362020*
         02362270  02362275  02362280

L4
  0081 *02802100*
         02802300 *02900100*

L5
  0006 *00422140*
         00422180

L5
  0034 *00915200*
        *00916000* 00916000  00916020  00916080

L5
  0081 *02802100*
         02802300 *02910100*

L6
  0006 *00422140*
         00422180

L6
  0029 *00723470*
        *00723650* 00723730

L6
  0034 *00915200*
        *00916260* 00916420

L6
  0081 *02802100*
         02802300 *02929100*

L7
```

```
  0006 *00422140*
         00422180

L7
  0081 *02802100*
         02802300 *02930100*

L8
  0006 *00422140*
         00422180

L8
  0081 *02802100*
         02802300 *02938100*

L9
  0081 *02802100*
         02802300 *02942100*

M
  0006 *00647000*
         00647000  00648000

M
  0007 *00433100*
         00433400

M
  0006 *01599000*
         01601000  01610000

M
  0006 *01625000*
         01627000  01628000  01634000  01639000  01646000  01647000  01651000

M
  0034 *00915140*
         00915840  00915860 *00915880* 00915880  00916100 *00916100*
```

```
M
  0080 *02333100*
         02333120  02333360

M
  0084 *02524000*
         02524000  02526000

MAKEINT
  0077 *02158000*
         02163000  02164000  02168000  02230120

MAKEINT
  0090 *03089000*
         03094000  03095000  03150000

MAP
  0002 *00191000*
         03390000  03391000  04209000

MATH
  0121 *04051000*
         04059000  04321000  04330000  04337000  04340000

MAX
  0002 *00188000*
         02754000  02797100  02870000 *03563910* 03563940 *03563940* 03563940  03563950 *03571000*

MAXCOM
  0002 *00214000*
         00215000  03571100

MAXDOS
  0002 *00230000*
         00231000  05034000

MAXEL
```

```
  0002 *00156000*
       00157000  04484000

MAXFILES
  0002 *00209000*
      *02183000* 02183000  03360000 *04586000* 04586000

MAXNBHANG
  0002 *00240000*
       00241000  01614000  01620000  01656000

MAXOPFILES
  0002 *00207000*
       00208000  02180000

MAXSTRING
  0002 *00201000*
       00202000  02653000  02703000

MDESC
  0090 *03084000*
       03088000  03315000

MDS
  0002 *00323000*


MEMHANDLER
  0006 *05535010*
       05535610  05576000  05576100  05576200  05577000

MESSAGE
  0002 *00242000*
       00436000  00454000  00472000  00490000  00508000  00526000  00544000  00562000  00580000  00595000  00596501
       00596527  00598040  00598160

MFID
  0090 *03041000*
       03048000
```

MFID
0090 *02986000*

*03160000**03184054**03187000* 03203000 *03234000* 03238000 03239000 *03270000* 03274000 03275000

MINUS
0002 *00301000*

01829000 01926000 01935000 02153000 02947000 04065000 04390000 05529470

MINUSP
0134 *04656500*

04732500 *04733000**04736000**04763500* 04763500

MKABS
0078 *02260000*

02260300 02260900 02263000

MKABS
0080 *02333540*

02333550 02371002 02372040 02382200 02404200 02414280 02419200

MKABS
0097 *03370000*

03371000 03375000 03376000 03378000 03378100 03379000

MKABS
0090 *03037000*

03039000 03040000 03127000

MKS
0002 *00324000*

00808000 01591000 01709000 03154000 03283000 03284000 04053000 04118000 04362000 04415011 04418000
04619030 04820000 04939000 05524000 05535150 05535500

ML2
0002 *00329000*

04310000 04314000 04317000

MNM

```
  0043 *01193000*
       *01209000**01220000**01228000* 01233000

MODEC
  0090 *03025000*
        03318000

MODEF
  0090 *03024000*


MORETHAN6
  0084 *02552000*
        02555000  02556000  02562000

MOVE
  0075 *02042000*
        02048000  02051000

MOVE
  0080 *02334000*
        02337000  02364240  02364830  02420070  02426040

MOVE
  0090 *03062000*
        03065000  03343000

MOVEANDBLOCK
  0090 *03066000*
       *03074000* 03083000  03348000  03351000  03363700

MOVEBACK
  0075 *02049000*
        02054000

MOVEC
  0080 *02337100*
        02364850
```

MOVESFQ
  0006 *00422010*

        00599500 02380200 02380300 02401200 02401300 02420011 02420050 02420060 02420120 02426030

MOVEW
  0006 *00647000*

        01648000 01845000 02260925 02343000 04938000 05025000 05529215

MOVEW
  0078 *02260925*

        02262800

MOVEW
  0080 *02343000*

        02362000 02380100 02395000 02401100 02414200 02415000

MSFL
  0002 *00242050*

        00434010 00434030

MSRW
  0002 *00241900*

        00242000 00242050

MUL
  0002 *00324000*

        00718000 00720000 03681000 03696000 03708000 03712000 04028040 04306000 04311000 04318000 04318010
        04319000 04320000 04355000 04359000 05343000

MULTI
  0140 *04788000*

        04791000 04793000 *04795000*

M1
  0036 *00944000*

        01012000 01021700

M2

```
  0092 *03070000*
         03070000  03077000

M30
  0037 *01030000*
         01033000  01046000

N
  0006 *00413000*


N
  0006 *00422255*
         00422255  00422268

N
  0006 *00394000*


N
  0007 *00433100*
         00433100  00433200  00600020  00600030  00636000  00641000  00646100

N
  0006 *00410000*


N
  0006 *00615000*
         00621000

N
  0006 *00608000*
         00613000

N
  0006 *00422130*
         00422190  00422200
```

```
N
  0006 *00411000*


N
  0006 *00422270*
         00422280

N
  0006 *00406000*


N
  0006 *00422220*
         00422220  00422230

N
  0007 *00426000*
         00426000  00428000

N
  0006 *00405000*


N
  0006 *00409000*


N
  0006 *00422100*
         00422110

N
  0006 *00423000*
         00434010  00434030  00434040  00598040  00598050  00598060  00598160

N
```

```
  0006 *00601000*
       00606000

N
  0006 *00408100*


N
  0006 *01261100*
       01261100  01261120  01261140

N
  0006 *01378000*
       01381000  01383000  01384000  01394000  01396000  01398000  01400000  01400400  01401000

N
  0006 *01235000*
       01249000

N
  0006 *01558000*
       01560000 *01567000* 01567000  01568000

N
  0006 *01353000*
       01356000 *01356000* 01357000

N
  0006 *01582000*
       01586000 *01586000* 01586000  01588000  01588100  01588400

N
  0006 *01428000*
       01432000  01435000

N
  0006 *01570000*
       01572000  01579000 *01579000* 01580000
```

```
N
  0006 *01267000*
       *01270000* 01270000  01271000

N
  0006 *01993000*
        01997000  01998000  02013000  02014000  02015000

N
  0006 *04774000*
        04812020  04812050  04814000  04825100  04840000  04842000  04844000

N
  0006 *04619010*
        04619040

N
  0006 *05535010*
        05535050  05535070 *05535070* 05535150  05535200
```

```
N
  0030 *00727000*
       *00750000* 00751000  00755000

N
  0045 *01197000*
       *01199000* 01200000  01201000 *01201000*

N
  0052 *01406000*
       *01418000**01419000**01420000**01421000* 01423000  01426400

N
  0071 *01963000*
       *01984000**01985000* 01987000  01987100  01988000  01989000

N
```

0080 *02362855*

02362855 02362865

N
0080 *02333100*

02333120 02333280

N
0081 *02513000*

02842000 02843000 02845000 02846000 *02917000* 02919000 02923000 02924000

N
0084 *02523100*

02523100 02523300

N
0084 *02524000*

02524000 02527000

N
0084 *02540000*

02540000 02541000

N
0097 *03372000*

03372000 03373000 03573000 03632000 03633000 04051000

N
0112 *03574000*

03584000 03585000 03586000 *03590000**03592000**03593000* 03596000 *03596100**03600000**03602000* 03605000
03607100 03607100 03609000 03611000 03615000 03618000

NAME
0002 *00150000*

00915220 00915280 00915340 01913100 01913300 *02721000**02974000* 03618100 03863000 03864000 03868000
04060000 04406030 04407000 04415000 04415310 04415320 04417000 04569000 04792000 04794000 04795000
04800100 04801000 04810000 04861000 04868000 04970000 04987100 05070000 05107040 05108000 05108100
05117100 05177000 05177100 05243200 05297000 05297300 05307000 05317000 05327000 05375000 05380000
05385000 05442000 05442500 05448500 05474000 05529440 05529450 05529500 05529510

NAME
  0034 *00915340*

        00915560 00915580 00915600 00915620 00915680 00915700 00915720 00915760 00915780 00915800 00915960
        00916000 00916020 00916080 00916160

NAMEDFSC
  0002 *00169080*

        04493150 04493300 04883300 04956010 04957050 04957100

NAMEIND
  0002 *00199130*

        00691530 *00692020* 01071700 *01079000* 01087000 01088000 *01088000**01096000* 01101000 01111000 01113000
        01116000 01117000 03649250 04500725 *04500725* 04500750 04503175 04516200 04956040 *04956600*

NAMEL
  0006 *05430000*

        05466000 05562000

NAMELIST
  0002 *00279000*

        00760000 04890000 04900000 05437000 05438000 05442500

NAMETOG
  0139 *04785050*

       *04890100**04900100* 04950000 04954050 04956750

NAME1
  0006 *00600030*

         00600050 00600080

NAME2
  0006 *00600030*

         00600060 00600080

NAMLIST
  0002 *00199200*

        03649230 03649250 04500750 04503170 04503175 04726000 04739500 04740500 04956040 04956500 04956600

NCR

0002 *00176000*

02428000 02502000 02646000 *02657000* 02657000 02663000 *02663000* 02669000 *02669000**02706000* 02706000
02714000 02714000 02814000 02944000 04680000 *04680000* 04681000 *04681000* 04687650 *04687650**04689000*
04689000 04708520 *04708520**04708530* 04708530 04730120 *04730120* 04730140 *04730140* 04737000 *04737000*
05067390 05067390 05529400 *05529400*

NCR
0081 *02514000*

02520000

NCR
0081 *02448000*

02483000

NCRV
0081 *02448000*

02449000 02455000 02456000 02457000 02496000 02497000

NCRV
0081 *02514000*

02514000 02516000 02518000 02520000

NCRV
0134 *04657500*

04657500 04658000

NCRV
0134 *04676000*

04676000 04676500 04774000 04787000 04959000 05063600 05100000 05101000 05479000 05535010

NEED
0006 *00401000*

01594000 01976000 *03573000**03618000* 03619000 03649300 03990104 03996000 04055000 04117000 04364000
04415300 04417000 04503190 04834000 04943000 04954000 04954050 04954100 04954150 04954200 04956700
04956750 04956800 04956925 04956930 05108000 05168000 05174000 05177000 05297000 05535160 05535520

NEQL
0002 *00324000*

04229000 04245000 05316600 05395000 05407000

NEW
  0006 *00422010*

        00422010 00422100 00422130

NEW
  0080 *02362300*

        02362500

NEWSEQ
  0006 *02317000*

        02320000 02364560 02364605 02364660 02364680 02364830 02374053 02389400 02409400 02420057 02423000

NEWTAPE
  0006 *00388000*

        02364860 02374060 02374065 02426005 03366200

NEWTPTOG
  0002 *00169068*

        02269900 02275100 02364330 02364810 02364811 02374055 02374110 02374300 02374580 02374680 02426000

NEWWD
  0134 *04657100*

        04747400 04749000 04749600

NEXT
  0002 *00150000*

        00363000 00623000 00625000 01018110 01829000 01830000 01856000 01896000 01914000 01926000 01927000
        01933000 01935000 01936000 01940000 01943000 01944000 01948000 01949000 02153000 02155000 02167000
        02191000 02193000 02196000 02199000 02206000 02230090 02240000 02241000 *02575000**02588000**02595000*
       *02625000**02720000**02748000**02804000**02827000**02828000**02830000**02843000**02844000**02849000**02857000*
        02860000 *02862200**02864000**02901000**02919000**02922000**02925100* 02925100 *02930000**02943000**02945000*
       *02946000**02947000**02948000**02949000**02952000**02961000**02964000* 02972000 03668000 03673000 03801000
        03808000 03819000 03820000 03864000 03867000 03947000 03949000 03981000 03982000 03990136 03990170
        03990180 04064000 04065000 04073000 *04076100* 04076100 04077000 04079000 04090000 04114200 *04114300*
        04124100 *04124200* 04129100 *04129200* 04133000 04174000 *04184100* 04184100 04185000 04189000 04194000
        04203000 04389000 04390000 04404200 04404250 04404650 04415100 04415110 04415130 04420000 04493000
        04493100 04494000 04497000 04502000 04524000 04530000 04533000 04547000 04555000 04561000 04562000
        04563000 04564000 04566000 04571000 04573000 04576000 04577000 *04770000* 04799000 04800000 04802000
        04803000 04813000 04840000 04842000 04852000 04857000 04858000 04860000 04866200 04883000 04883100
        04883150 04883200 04883500 04883650 04906000 04908000 04909000 04912200 04912300 04914000 04915000

04916000 04916100 04917000 04918000 04930000 04944000 04969000 04972000 04973000 04975000 05014000
05027000 05028010 05029200 05030000 05038000 05039000 05049000 05051100 05067210 05067240 05068100
05071300 05072000 05097000 05098000 05106000 05110000 05115000 05116000 05117100 05119000 05120000
05141000 05144000 05166000 05172000 05174000 05176000 05179000 05197000 05237000 05243000 05247000
05256000 05261000 05262000 05281000 05282000 05287000 05296000 05299000 05300220 05300260 05300290
05300300 05305000 05311000 05313600 05314300 05315600 05317600 05318000 05321000 05326000 05329000
05330000 05332000 05357000 05360000 05374000 05377000 05379000 05382000 05384000 05434000 05435000
05443000 05444000 05459000 05460000 05464000 05471000 05472000 05480020 05493000 05500000 05529220
05529270 05529290 05529420 05529470 05529610 05529620 05529630 05529635 05529820 05535090 05535180
05535225 05535450 05535595 05538000 05539000 05540000 05578000 *05578000* 05595000 05598000 05612000

NEXTACC
0002 *00177000*

02577000 *02586000* 02587000 02597000 02600000 02601000 02831000 02850000 *02851000*

NEXTACC2
0002 *00177000*

02586000 02587000 02851000

NEXTCARD
0002 *00174000*

02364160 *02364900* 02365000 *02367000**02371000**02372000* 02372010 *02372030* 02373010 *02373010**02373030*
*02374200**02374600* 02376000 02378000 *02387000* 02393000 02399000 02402000 *02407000**02408500**02411000*
02414100 *02416200**02417000* 02418100 02420010 02491650 02491700 02491800 02492000 02492400 02492500
02492600 *03380000* 04667600 04668000 04668500 04668600 04669000 04670000 04670500

NEXTCOM
0002 *00216000*

00646600 *00646600* 00646700 00740000 *00786000**01756000* 05088000 05185000 *05185000* 05186100 05191000
05195000 *05238000* 05238000 05238200 05271000 05286000

NEXTEXTRA
0002 *00223000*

03387000 03954000 03958000 03959000 *03959000* 03990108 03990116 03990118 *03990118* 04009000 04014000
04016000 04016000 04475000 04479000 04481000 *04481000*

NEXTINFO
0002 *00223000*

00746000 *00791000**01755000* 01760000 01762000 *01766000* 01766000 01770000 03990128 *03990128* 03990128
04989000

NEXTRA
0002 *00258000*

00711000 00815000 03232000 03649000 03859000 03963000 03985000 04424000 04458000 04460000 04503100
05262200 05451000 05529730

NEXTSCN
0002 *00174000*

*02585000**02598000* 02831000

NEXTSS
0002 *00224000*

00787000 03799000 03815000 *03816000* 03816000 04435000 04438000 04439000 *04439000*

NEXT1
0084 *02531000*

02534000

NEXT2
0084 *02531000*

02537000

NF
0139 *04784000*

04857000 *04863000*

NIM
0155 *05431000*

05435000 05464000

NLABELS
0131 *04431000*

04449000 *04449000* 04451000 04452000

NNEW
0002 *00159000*

03978000 *04060000* 04433000 04603000 *05108000*

NOFND
0134 *04657000*

04715000 04715600 04716000 04716500 04717000 04717500 04721000 04729500 04730500 04731500 04746100
04746750 *04762000*

NOFORM
  0139 *04784000*

       04864000 *04917000*

NOFORMT
  0139 *04785020*

       04863500 04954150 04956925

NOLIN
  0002 *00242300*

       00794400 01021360 01021500 01021600 01021950 02432200 *02432200* 02432300

NOP
  0002 *00324000*

       01377000 01400600 01426600 01534100 01701000 01795100 01808000 01811100 04877000 04894000 04903000
       05349300

NOPLACE
  0134 *04657000*

       04681002 *04767500*

NOSCAN
  0121 *04021000*

       04069000 *04072000* 04390000

NOTOP,0
  0002 *00169075*

       04584000 04617000 04619050 04619080

NOTZEROP
  0115 *03850000*

       03859000 03944000

NOWRI
  0080 *02362902*

       02374050 02425100

NPARMS

```
  0032 *00798000*
       *00815000* 00816000  00818000  00819000  00820000  00904000  00905000

NPARMS
  0115 *03847000*
       *03948000* 03953000  03963000  03964000

NPARMS
  0117 *03971000*
       *03980000* 03980000  03985000

NPARMS
  0120 *03993000*
       *04007000* 04009000  04010000 *04010000* 04012000 .

NPARMS
  0141 *04963000*
       *04970000* 04970000  04980000  05000000  05003000

NSFG
  0002 *00162000*
        00629000 *00780000* 00781000  00788000  00932000  00932700  00933000  00953000  00963000  00964000  00972000
        00975000  00980000  00984000  00991000  00995500  00999100 *01001000* 01002000  01028000  01542000  01699000
        01799000  02001000  02179000 *03106200**03107000* 03130000  03288000  03289000 *03300000* 03654000  03657000
        03663000  04415050  04415090  04434000  04527000  04549000  04773070  04773140  04773160  04816136  04921000
        04923000  04998000  05041400  05041700  05048000  05215000

NSFG
  0037 *01028000*
       *01035000* 01036000

NSFGS
  0092 *03069000*
       *03071000* 03072000

NSUBS
  0028 *00709000*
       *00711000* 00713000

NSUBS
```

0114 *03793000*

03812000 03817000 *03817000* 03830000 03836000

NSUBS
0113 *03634000*

03649000 03649200 03649230 03649250 03650000 03652000 03666000 03677000 03685000

NSUBS
0132 *04487100*

04503100 04503160 04503170 04503175 04509100

NTAPTOG
0002 *00169099*

02275100 03366200

NUB
0006 *02434000*

02443000

NUL
0134 *04657000*

04685500 04697500 04726540 *04764500*

NUL1
0134 *04657000*

04686500 04687655 04734500 04742000 04752000 *04760500*

NUM
0002 *00301000*

01830000 01927000 01936000 02625000 02720000 02748000 02843000 02864000 02919000 02925100 03808000
03867000 04077000 04174000 04185000 04657000 04800000 04866200 05039000 05067240 05141000 05256000
05305000 05315600 05326000 05374000 05379000 05384000 05472000 05529290

NUM
0134 *04657000*

04708500 04713000 04713500 04714000

NUMBER
0081 *02801000*

NUMCLASS
0002 *00291000*

03657000 03663000 03874000 04183000 04194000 04198100 04201000 04602500 05521100

NUMFINI
0088 *02729000*

02765000 *02785000*

NUMINTM1
0002 *00155210*

00233000 03219000 03997000 05130100

NUMTYPE
0002 *00150000*

01832000 01851000 *01851300* 01853000 01927000 01936000 *02626000**02645000**02675000**02722000* 02733500
*02733500**02749000* 02749000 *02757000**02779500* 02779500 *02780000* 02789000 *02864000**02873000**02912000*
02920000 03810000 04176000 *04177400**04177800* 04179000 04181000 05067240 05256000 05529290

NXAVIL
0002 *00182000*

00780000 00780000 *01632000* 01632000 01633000 01642000 *01642000* 01643000 01669000 02993000 03104000
03104000 03105000 *03106200* 03106200 03106210 03107000 *03107000* 03108000 03111000 *03111000* 03112000
03123000 03123000 03124000 03239000 *03239000**03259000* 03259000 03260000 03263000 03275000 *03275000*
03300000 03300000 03304000 03351000 03352000 03362000 03363400 03363700 03888000 *03888000* 03890000
04934000 04934000 04935000 05619010

NXT
0030 *00727000*

00749000 00750000 00756000 00759000 00761000 *00774000*

NX1
0002 *00163000*

00691550 *00692040* 01072200 01082000 01083000 01084000 01085000 01091000 01095000 01097000 01099000
01100000 01111000 01112000 *04801000**04838000* 04952000 04956910 *05375000* 05390000 05397000 05403000
05411000

NX2
0002 *00163000*

*04801000**04838000* 04953000 04956920 *05380000* 05391200 05396000 05403200 05414000

```
NX3
  0002 *00163000*

       *05385000* 05391000  05397200  05402000  05410000

OFLOWHANGERS
  0006 *01612000*

        01623000  04871000  04901000

OK1
  0081 *02497000*

        02509000

OK2
  0081 *02497000*

        02509000

OLn
  0006 *00422010*

        00422010  00422100  00422140

OLD
  0080 *02362300*

        02362650

OP
  0002 *00235000*

       *02845000**02899000**02900000**02901000**02923000**02931000**02945000**02946000**02947000**02950000**02952000*
       *02963000* 04028030 *04066000**04076100**04114300**04124200**04129200**04184100* 04384000 *04391000*

OP
  0006 *00412000*


OP
  0006 *00723100*


OP
```

```
  0006 *01365000*
       01374000

OP
  0006 *01359000*
       01362000

OPHASE
  0134 *04647000*
       04731500

OPST
  0002 *00196000*
       04063000  04210000  04384000

OPTYPF
  0118 *03990045*
       03990200

OPTYPF
  0121 *04026000*
       04058000  04062000  04075000  04123000  04181000  04192000  04196000  04200000  04209000  04250000  04258000
       04260020  04279000  04291015  04308000  04328000  04348000  04367000  04368000  04369000  04372000  04397000
       04398000

OWNID
  0041 *01129100*
       *01129200* 01159000  01163100  01165100

OWNID
  0063 *01706100*
       *01708100* 01712100  01719100  01720100  01725000  01728000  01733100

P
  0002 *00151000*
       01071700 *01440000* 01518000  02140000  04773020  05232000 *05322000* 05327000 *05327000* 05334000  05339000
       05345000  05349000

P
```

0007 *00433100*

00433100 00433400

P

0006 *00422255*

00422260

P

0006 *00422220*

00422220 00422230

P

0006 *01273000*

01276000 01280000 01287000

P

0006 *01303000*

01306000 01311000 01325000 01331000

P

0006 *02434000*

02434000 02437000 02443000

P

0040 *01071700*

01079000 01087000 01088000 01096000 01101000 01111000 01113000 01116000 01117000

P

0075 *02042000*

02044000

P

0075 *02049000*

02050000

P

0078 *02260925*

02260950

```
P
  0077 *02140000*
       *02186000**02187000* 02187000  02188000  02190000  02243000

P
  0080 *02342010*
        02342010  02342020  02342030

P
  0080 *02334000*
        02335000  02336000

P
  0084 *02552000*
        02553000

P
  0097 *03370000*
        03371000
```

```
P
  0138 *04773020*
       *04773080* 04773100  04773110 *04773140* 04773140

P
  0151 *05232000*
       *05251000* 05257000  05258000 *05258000*

PACK
  0006 *01262000*
        01266000  01270000  01356000  01362000  01394000  01396000  01398000  01400000  01423000  01432000  01567000
        01579000  01662500  01663000  01666000

PARAM
  0141 *04961000*
        04970000  04982000  04984000  04987000

PARAMETERS
```

```
  0006 *03842000*
        03968000  04120000  04421000

PARENCT
  0134 *04656000*
        04726000 *04726000* 04726500  04738500  04739000  04740000 *04740000* 04740100  04765000  04772500

PARMLINK
  0002 *00197000*
        03971000  04438000  04440000  04464000  04466000  04467000  04480000  04963000  05115000  05116000  05529740


PARMLINK
  0117 *03971000*
       *03972000* 03976000  03979000 *03979000*

PARMLINK
  0141 *04963000*
       *04989000* 05001000  05002000 *05002000**05004000* 05004000

PARMS
  0002 *00161000*
        00691510 *00692010* 01071100 *01087000**01089000* 01091000  04436000  04458000 *04460000* 04463000  04473000
        04474000  04477000 *05104000**05115000* 05115000 *05116000* 05116000

PARMSREQ
  0006 *05101000*
        05111000

PARMTYPE
  0115 *03852000*
        03893000  03958000

PAUSE
  0006 *05467000*
        05478000  05563000

PCOMM
  0151 *05232050*
```

```
        *05269000* 05270100  05270200

PDC
  0090 *03036000*

        03310000

PDIC
  0002 *00357000*

        01006000  01038000  01059000  03236000  03272000  03301000

PDINX
  0002 *00336000*

        00356000  00357000  01006000 *01018000* 01018000  01038000 *01042000* 01042000  01059000  01064000 *01064000*
        03106215  03236000 *03241000* 03241000  03272000  03277000 *03277000* 03301000 *03306000* 03306000  03309000
       *03381100*

PDTR
  0002 *00356000*

        01006000  01038000  01059000  03236000  03272000  03301000

PDPRT
  0002 *00335000*

        01006000  01038000  01059000  03106215  03236000  03272000  03301000  03308100  03310000  04404275  04404300

PDR
  0090 *03035000*

        03310000

PERIODWORD
  0002 *00190000*

        02745000  03434000

PF
  0134 *04656500*

        04710000  04722000 *04722500* 04735730  04757420  04757430  04758000 *04760900*

PLACE
  0007 *00426000*

        00433000  00598160
```

PLINK
  0032 *00799000*

       00822000 00825000 00833000 00834000 00899000 *00907000* 00908000 00909000 00910000

PLINKX
  0032 *00799000*

       00818000 00822000 *00901000* 00901000 00904000 *00904000* 00907000 00911000 *00911000*

PLUS
  0002 *00301000*

        01829000 01926000 01935000 02901000 04064000 04389000

PLUSP
  0134 *04656600*

       *04714600**04733000* 04760500 *04760900*

PNT
  0006 *00723140*

        00723160 00723200 00723210 00723250 00938000 00940000 01026000

POS
  0006 *00723140*

        00723150 00723255

POSIT
  0134 *04673500*

        04673500 04674500 05100000 05101000 05479000

POSTWRAPUP
  0006 *00393100*

        00606500 00613500 02281600 *05613900*

PPHASE
  0134 *04647500*

        04732000

PR

```
  0002 *00196000*

       04063000  04207000  04383000


PREC
  0002 *00235000*

       *02804000**02846000**02899000**02900000**02901000**02924000**02931000**02945000**02946000**02947000**02950000*
       *02952000**02963000* 03673200 *04066000* 04067000 *04076100* 04078100 *04114300**04124200**04129200**04184100*
        04184200  04207000  04383000  04386000 *04391000*


PRGDESCBLDR
  0006 *00416000*

       00932700  00933000  00953000  00964000  00972000  00991000 *01053000**01067000* 01068000  01543000  01589010
       01632000  01642000  01669000  01801000  02001000  03104000  03106200  03107000  03112000  03123000  03243000
       03247000  03259000  03279000  03288000  03299000  03888000  04415050  04773160  04921000  04935000


PRI
  0090 *02986000*

       *03147000* 03151000  03168000  03187500  03193000  03204000 *03216000**03222000* 03223000  03224000 *03226000*
        03226000  03230000  03234000  03245000  03247000  03248000 *03266000* 03267000 *03268000* 03268000  03270000
        03279000  03280000 *03280000*


PRINTBUFF
  0002 *00169300*

       00422160  00422180  00422190  00422200  00422280  00422285  00723520  00723640  00723650  00723670  00723720
       00916240  00916260  00916300  00916400  00916460  01018320  02374060  02380230  02380240  02389500  02389600
       02401230  02401240  02409500  02409600  02426000  02426005


PRINTCARD
  0006 *00400000*

       00422275  00604000  00611000 *02040000* 02055000  02083000  02086500  02088000  02093000  02095000  02177000
       02287000  02289500  02364890  02420100  02493200  02506010  02662000  02713000  02818010  04671500  04769500


PRL
  0002 *00324000*


PROD
  0113 *03636000*

       03692000 *03692000* 03696000 *03698000* 03707000  03712000


PRT
```

```
  0006 *01599000*
         01606000  01610000

PRT
  0006 *01055000*
         01058000 *01058000* 01061000  01065000  01067000

PRT
  0063 *01706000*
        *01707000* 01708100  01710000

PRT
  0090 *02979000*
         03308100  03320000  03348000

PRT
  0090 *03084000*
         03087000

PRTAC
  0002 *00347000*
         01061000

PRTADR
  0090 *02988000*
        *03312000* 03320000  03321000

PRTAF
  0002 *00346000*
         03312000  03314000

PRTLINKC
  0090 *03003000*


PRTLINKF
  0090 *03002000*
         03321000
```

```
PRTOG
  0002 *00169067*

        00734000  00755000  00766000  02270200  02272300  02275900  02276500  03204000  03209000  03244000  03251000


PRTS
  0002 *00161000*

        00396600  00396610  00398000  00689100 *00689100* 00689100  00691510 *00692010* 00932300 *01058000* 01058000
       *01058000* 01058000  01071200  01080000  01081000 *01081000* 01082000  01087000  01089000  01093000 *01094000*
        01094000  01095000  01096000  01119000  01121000 *01121000* 01121000  01122000  01122550 *01122600**01163100*
        01163100 *01163100**01165100* 01165100 *01165100* 01169000 *01169000* 01169000 *01169000* 01590000 *01590000*
       *01636000* 01636000 *01636000**01639000* 01639000 *01639000**01644000* 01644000 *01644000* 01644000  01657000
       *01657000**01662000* 01662000 *01662000* 01662300  01663000  01669000 *01679000* 01679000  01680000  01696000
       *01696000* 01697000 *02182000* 02182000 *02182000* 02187000  03264000  03348000  03350000  03357000 *03382000*
        04138000 *04138000* 04586500 *04586500* 04587000  04607000 *04607000* 04607000  04816135 *04816135**04932000*
        04932000 *04932000* 04936000 *04936000* 05619000


PRTSAVER
  0006 *01624000*

        01652000  04772600  04956040  05463000


PRTX
  0040 *01071800*

        01076000  01108000


PT
  0006 *00723450*

        00723800  01018330


PTR
  0006 *00391000*

        01018130  01018140  01018150  01018170  01018200  01018210  01018375  01019000


PTYPE
  0032 *00800000*

       *00823000* 00835000 *00836000* 00837000


PTYPE
  0115 *03847000*

       *03869000**03872000* 03875000 *03877000* 03893000  03901000  03903200  03904100  03928000  03930300  03930700
```

```
        03934000  03936100  03938000  03940000  03941000  03943100

PTYPEX
  0032 *00800000*

       *00819000* 00823000  00835000  00867000  00895000  00897000  00902000 *00902000*

PURGEINFO
  0006 *01751000*

        01760000  01770000

PUT
  0006 *00636000*

        00639000  00670000  00672000  00696100  00702100  00737000  00739000  00739200  00744000  01085000  01112000
        01114000  01134000  01135000  01158000  01171000  01590000  01610000  01617000  01651000  01680000  01795000
        01802000  01990000  02188000  02243000  03106150  03586000  03605000  03607100  03838000  03909000  03915000
        03955000  03990108  04009000  04413000  04423000  04434000  04448000  04476000  04773140  04876000  04893000
        04900000  04902000  04998000  05000000  05004000  05017020  05017030  05069200  05079400  05080000  05080500
        05093000  05107100  05132000  05133000  05196000  05227000  05228000  05270500  05276000  05437000  05535240

PUTC
  0006 *00646100*

        01075000  05091000  05186200  05238300  05274000  05284000

PWI
  0002 *00362000*

        01087000  01089000  05063670

PWROOT
  0002 *00362000*

        05090000  05191100  05195000  05273000  05286000

PXREF
  0002 *00169096*

        01018100  01018355  02282900

P1
  0006 *00422255*

        00422255  00422262  00422270

P2
```

0006 *00422255*

00422255 00422264

P3

0006 *00422255*

00422255 00422266

Q

0005 *00365000*

00365000 00367000 00370000

Q

0005 *00372000*

00372000 00373000 00394000 00394500 00399000 00400010 00401000 00402000 00403000 00404000 00405000
00406000 00408000 00408100 00409000 00410000 00411000 00412000 00413000 00414000 00415000 00416000
00418000 00421000 00422000

Q

0006 *00723140*

00723150 00723200

Q

0040 *01071400*

$0_1$083000 $0_1$084000 $0_1$088000 $0_1$097000 $0_1$116000

Q

0075 *02042000*

02042000 02044000

Q

0078 *02260925*

02260925 02260950

Q

0080 *02334000*

02334000 02335000

Q

0151 *05232000*

```
        *05239000* 05277000 *05277000* 05284000

QF
  0134 *04656500*

         04688000 *04689500**04696000*

QQQDISKDEFAULT
  0090 *02997010*

        *03164045* 03184060

QT
  0087 *02686000*

         02690000  02691000  02692000  02696000

QUOTESTRING
  0081 *02679000*

         02725000  02966000

R
  0006 *00403000*
```

```
R
  0006 *00723100*


R
  0006 *01359000*

       *01362000* 01362000  01363000

R
  0006 *01069000*

         01074000  01075000  01077000  01078000  01079000  01080000  01093000  01119000  01122000  01122550 *01122550*
         01122600 *01122600* 01122600

R
  0023 *00651000*

         00658000  00674000
```

```
R
  0025 *00706200*
       *00706340**00706360* 00706360  00706380

R
  0027 *00706733*
       *00706780* 00706800 *00706800* 00706800

R
  0023 *00650000*
        00687000 *00687000* 00691000  00691900  00691950  00697100  00704000

R
  0080 *02343000*
        02343000  02360000

R
  0080 *02362010*
        02362010  02362033
```

```
R
  0157 *05529120*
       *05529210* 05529630

R
  0151 *05232000*
       *05251000* 05258000  05259000 *05259000**05262200* 05262200  05262300

RANDOMTOG
  0002 *00169091*
        04076050  04114200  04124200  04129100  04184050  04850000  04850100

RATA
  0140 *04788000*
        04797000  04799000  04800000  04803000 *04810000*

RDTRIN
  0139 *04785000*
       *04839000* 04848000  04855000  04856000  04863000  04883000  04912100  04916000  04951000  04957050
```

RDV
  0002 *00325000*

        03990260 03990420

READACARD
  0006 *02056000*

        02080000 *02332000**02429000* 02433000 02493200 02505000 02661000 02712000 02817000 04671500 04769000

REALID
  0002 *00165000*

        01019160 03565000 05041100 05529220

REALS
  0006 *05484500*

        05568000

REALTYPE
  0002 *00297000*

        00896000 01452000 01469000 01483000 01927000 01936000 01988000 02626000 02733500 02749000 02779500
        02912000 03565000 03655000 03804000 03938000 03941000 03977000 03990160 04177800 04196000 04198000
        04258000 04279000 04294000 04308000 04328000 04348000 04367500 04536000 04544000 04550000 04556000
        04602000 04854000 04883550 04912400 04985000 04993530 05033000 05041000 05041800 05052000 05067320
        05067370 05300240 05335000 05484500 05521000 05568000

REC
  0006 *00723450*

        00723610 00723670 00723680 00723690 00723720

RECORD
  0077 *02146000*

        02232000 02236010 02237012 02237020 02237040 *02237050* 02238000

REL
  0024 *00653000*

        00669000 00670000

REL
  0040 *01071100*

01087000 01089000 01091000

RELADC
0002 *00349000*

01062000

RELADD
0002 *00260000*

01087000 01089000 01096000

RELADF
0002 *00348000*

03315000

RELADR
0006 *01055000*

01062000 01065000

RELATION
0121 *04029000*

04224000 04225000 04226000 04227000 04228000 *04230000* 04246000

REMFIXED
0002 *00169089*

02273500 02273800 02431005

REMOTE
0006 *00389500*

00598080

REMOTETOG
0002 *00169090*

02273500 02273800 03385355 04841100 04845100

REPEAT
0134 *04655500*

04684500 *04684500* 04687590 *04709500* 04718000 04719500 04725550 *04725550* 04726500 *04727500* 04732500
04735720 04746000 04747000 04747400 04749000 04749200 04749600 04749800 *04758400* 04758400 04759000
*04760980**04762500* 04762510

```
REPLACEMENT
  0077 *02165000*

       *02168000* 02171000  02223000  02230030  02230045  02231000  02232000  02233000

RESERVED
  0002 *00292000*



RESERVEDWORDS
  0002 *00195010*

       02591000  02592000  03402000

RESERVEDWORDSLP
  0002 *00195000*

       02571000  02572000  03399000

RESET
  0078 *02261800*

       *02268400* 02273200

RESLENGTH
  0002 *00195010*

       02591100  02597000  03412000

RESLENGTHLP
  0002 *00195000*

       02571100  02577000  03408000

RESULT
  0002 *00174200*

       02364380  02364381  02364382  02364385  02364400  02364420  02364440  02364460  02364480  02364500  02364540
       02364580  02364600  02364620  02364640  02371004  02372050  02382300  02404300  02414290  02419250

RESULT
  0080 *02333100*

       02333200

RESWD
```

```
  0084 *02523000*
        02561000  02562000 *02579000*

RETURN
  0006 *05495000*
        05529000  05569000

RETURN
  0141 *04963000*
       *04979000* 04994000

RETURNFOUND
  0002 *00169072*
        00792000  00949100  04408100  05486100  05497100

RINGCHECK
  0006 *05063100*
       *05063400* 05063500  05089100  05272100

RITE
  0006 *00391000*
        00422110  00422200  00723720  00916400  02287600  05617000  05618100  05619000  05619020  05619040  05619050

RLESSC
  0121 *04028050*
        04261000 *04282010*

ROOT
  0002 *00166000*
        00362000  05063400  05063670  05085000  05089000  05090000  05093000 *05186000**05186100* 05186200 *05190000*
        05191000  05191100  05195000  05196000 *05238100**05238200* 05238300  05268000  05272000  05273000  05276000
        05284000  05286000

ROUND
  0134 *04657000*
       *04681000* 04681120  04686000  04686500  04687530  04690500  04699000  04712000  04714600  04720000  04723000
        04730150  04735620  04735740  04736000  04743500  04744500  04767000  04769500

ROUND
```

0132 *04488000*

04492000 04567000 04573000

ROUND1
0134 *04657100*

04680700 04728000 04744500

ROW
0006 *00422100*

00422110

RP
0134 *04657000*

04724000 *04736500*

RPAREN
0002 *00302000*

01940000 02948000 03673000 03820000 03949000 03982000 03990180 04203000 04404650 04415130 04533000
04561000 04563000 04576000 04802000 04857000 04908000 04915000 04973000 05120000 05262000 05282000
05300260 05318000 05330000 05360000 05529620

RPLUS
0041 *01127000*

01140000 01141000 01142000 01145000 01147000 01148000 01149000 01150000 01151000 *01168000*

RPLUSD
0121 *04028050*

04260010 04281000 04310000

RPT
0070 *01817000*

01838000 01864000 *01880000* 01885000 *01885000* 01889000 *01891000* 01921000 01954000

RSH
0002 *00194040*

00195010 02591000

RSH1
0002 *00194040*

02592000 05539000

RSP
0002 *00194020*

00195000 02571000

RSP1
0002 *00194020*

02572000

RSTORF
0086 *02638000*

02642000 02643000 02663000

RTI
0002 *00152000*

02027000 02028000 *05610000* 05619020

RTN
0002 *00325000*

05507000

RTPARN
0134 *04649000*

04748800

RTS
0002 *00325000*

04415250 04415270 04521000 04926000

RWP
0002 *00194020*

00195000

RWS
0002 *00194040*

00195010

R1

```
  0157 *05529110*

       *05529220**05529340* 05529475  05529570 *05529720* 05529730 *05529730* 05529730  05529740

R2
  0157 *05529110*

       *05529440* 05529440  05529450  05529475  05529520  05529530  05529550 *05529550* 05529550 *05529560* 05529570
        05529730  05529740  05529760

R3
  0157 *05529110*

        05529220 *05529220* 05529230  05529270  05529310  05529340  05529350  05529390  05529500 *05529500* 05529500
        05529510  05529520  05529530 *05529680* 05529690  05529700 *05529740* 05529770  05529790

R4
  0157 *05529110*

       *05529430* 05529475  05529530 *05529760* 05529770  05529790

S
  0006 *00402000*

S
  0006 *00706560*

        00706570  00706600  00707000  00723100

S
  0006 *00723140*

        00723150  00723210

S
  0007 *00433100*

        00433300  00641000

S
  0007 *00426000*

        00430000

S
  0006 *01174000*
```

```
          01174000  01179000  01187000  01190000  01195000  01235000

S
  0006 *01126000*

       *01129200* 01129200  01130000  01134000  01135000  01158000  01159000  01162000  01171000

S
  0006 *01190000*

        01206000  01208000  01209000  01210000 *01210000* 01211000  01212000  01213000  01217000  01219000  01220000
        01227000  01229000

S
  0025 *00706200*

       *00706280* 00706360

S
  0043 *01195000*

        01200000  01202000

S
  0049 *01275000*

       *01276000* 01277000  01278000  01281000  01289000  01291000 *01291000* 01292000  01300000

S
  0048 *01255000*

        01257000

S
  0050 *01305000*

       *01306000* 01307000  01308000  01309000  01312000  01314000  01323000  01326000  01328000  01332000  01335000
        01336000  01338000  01347000  01348000  01350000  01351000

S
  0070 *01824000*

       *01829000* 01900000  01943000

S
  0078 *02260000*

        02260300  02260600
```

```
S
  0077 *02172000*
         02173000

S
  0080 *02333540*
         02333550

S
  0085 *02607000*
        *02620100**02621000* 02623000

S
  0085 *02615000*
         02615000  02617000

S
  0086 *02633000*
         02633000  02635000

S
  0087 *02683000*
         02684000  02687000  02697000  02698000

S
  0088 *02730000*
         02738000 *02738000**02773000* 02773000  02774000  02775000 *02778000* 02778100

S
  0086 *02638000*
         02639000  02642000

S
  0090 *03089000*
         03091000  03094000

S
```

0151 *05232000*

05252000 05258000

SADR
0062 *01655000*

01668000 01699000

SAVEADR
0113 *03637100*

03654000 03655500 03659000 03664100

SAVEADR
0121 *04025600*

04178500 04350000

SAVEBRAD
0141 *04962000*

04976000 04996000

SAVECARD
0002 *00174100*

02364160 02374200 02374600

SAVECODETOG
0002 *00169055*

SAVEDEBUGTOG
0002 *00169054*

SAVELABL
0154 *05354000*

05365100 05366100

SAVELISTOG
0002 *00169057*

02261500

SAVFLISTOG
0078 *02261500*

02262775 02289450

SAVENSEG
0113 *03637100*

03654000 03657000 03663000

SAVEPRTTOG
0002 *00169056*

SAVFR
0077 *02148000*

02223000 02236020 02239000

SAVESUBS
0002 *00199120*

00691530 *00692020* 01071600 *01091000* 01096000 03106100 03106150 *03649200* 03649200 *04503160* 04503160

SAVETIME
0002 *00146000*

00392000 03365000

SAVETOG
0002 *00169050*

00169053 00169054 00169055 00169056 00169057 00169058 00169059 00169060 00169061 00169062 00169063
00169064 00169065 00169066 00169067 00169068 00169069 00169070 00169071 00169072 00169073 00169074
00169075 00169076 00169077 00169078 00169079 00169080 00169081 00169082 00169083 00169084 00169085
00169086 00169087 00169088 00169089 00169090 00169091 00169092 00169093 00169094 00169095 00169096
00169097 00169098 00169099 00169100 00422110 00422190 *00422275* 00422275 00598000 00598050 00598120
00604000 00611000 *00691800* 00693500 *00693600**00706300**00723640* 00723690 *00723720* 00734000 00739100
00755000 00766000 00781000 00783000 *00792000**00794000* 00794400 *00916240* 00916340 *00916400* 00949100
01005000 01018025 01018100 01018110 01018130 *01018280**01018355* 01018370 01018375 01019000 *01020000*
01021350 01021957 01050000 *01085500**01661000**01702000* 01728000 *01730000**01826000**01960000**02034000*
02052000 02053000 02054010 02083000 02086000 02086500 02088000 02093000 02095000 02177000 02217000
02262775 *02263900**02264800**02269300**02269600**02269900**02270200**02270500**02272300**02272600**02273500*
02273500 *02273800* 02273800 *02274100**02274700**02275100**02275300**02275600* 02275600 *02275900**02276500*
*02276800**02277100**02277400**02277700**02279200**02280100**02282900**02283200**02285200**02285500**02285800*
*02286100**02286400* 02286700 *02287000* 02287000 02287300 02289450 02289500 *02289600**02363100**02364330*
02364810 02364811 02364820 02364860 02364890 02373045 *02373100**02373200* 02374053 02374055 02374110

```
      *02374110**02374300* 02374580 *02374580**02374680* 02380000  02380100  02380125  02380210  02380220 *02380240*
       02389200  02389300  02389400 *02389600* 02395000  02396100 *02396200* 02401000  02401100  02401125  02401210
       02401220 *02401240* 02409200  02409300  02409400 *02409600* 02414175  02414200  02414225  02415000  02416100
      *02416200* 02419000 *02419050**02420000* 02420015  02420020  02420053 *02420090* 02420100  02421000  02426000
       02426010 *02431000**02431005**02431100* 02431100  02432000  02432100  02491600  02492450  02492550  02493200
       02506010  02655000  02656100  02662000  02707000  02708100  02713000  02818010  02827100 *02827100* 02939000
      *03129000* 03139000  03187100  03204000  03209000  03244000  03251000  03363200  03366200 *03384000**03385000*
      *03385100**03385200* 03385350 *03385350**03385355* 03385355  03385360 *03385400* 03385400 *03569000**03571300*
       03894050  03894060  03894100  04076050  04114200  04124200  04129100  04184050  04406030 *04408100* 04415000
      *04492100* 04493150 *04493300* 04500100  04500700  04503125  04509050  04513000  04516100  04584000  04604060
       04617000 *04619050**04619080* 04667500  04669500  04670100  04671500  04681020  04681030  04767500  04768000
       04769500 *04815000* 04816000  04841100  04845100 *04850000**04850100**04883300* 04883650  04928000  04956010
       04957050 *04957100* 05032000  05067000  05213000  05233000  05292000 *05365000**05367000* 05433000  05467100
       05468100 *05486100**05497100* 05612100  05614000  05615000  05618100

SAVINS
  0062 *01655100*

      *01683100* 01684100 *01695100* 01696100

SAVIP
  0121 *04025000*

      *04063000* 04074100  04395000  04396000

SAVIT
  0115 *03851000*

      *03853000* 03948000  03956000  03967000

SAVIT
  0113 *03639000*

      *03644000* 03689000  03785000

SAVIT
  0121 *04025000*

      *04062000* 04208100  04399000

SAVLASTLP
  0134 *04655500*

      *04726000* 04738500

SAVTOTAL
  0134 *04656000*

      *04689500* 04692000
```

SBVEXP
0002 *00290000*

03922000 03927000

SB2
0002 *00329000*

04240000 04241000 04285000 04288000

SCAN
0006 *00420000*

00625000 00627000 00630000 01827000 01829000 01847000 01855000 01900000 01913500 01925000 01926000
01932000 01934000 01935000 01939000 01941000 01943000 01948000 02152000 02154000 02241000 02245600
02446000 02975000 03670000 03673200 03818000 03819000 03839000 03862000 03866000 03870000 03975000
03983000 03990140 03990180 04070000 04076000 04121000 04127000 04184000 04191000 04197000 04388000
04404200 04404250 04404650 04404660 04414000 04415090 04415110 04415120 04415300 04419000 04421000
04493300 04496000 04512000 04514000 04528000 04534000 04544000 04550000 04556000 04561000 04565000
04572000 04619030 04771000 04790000 04798000 04799000 04802000 04803000 04813000 04850000 04853000
04859000 04864000 04883400 04883550 04883600 04905000 04908000 04912000 04912200 04912400 04915000
04916000 04930000 04945000 04968000 04971000 04973000 04992000 05013000 05018000 05020000 05028000
05037000 05051000 05067230 05067390 05071000 05072000 05097000 05098000 05105000 05109000 05114000
05118000 05121000 05140000 05143000 05147000 05152000 05158000 05165000 05173000 05177200 05181000
05236000 05242000 05246000 05255000 05260000 05263000 05283000 05295000 05297500 05300184 05300230
05300280 05300300 05304000 05308000 05313300 05314000 05315300 05317300 05318200 05325000 05328000
05331000 05333000 05356000 05358000 05373000 05376000 05378000 05381000 05383000 05386000 05434000
05435000 05443000 05444000 05459000 05470000 05475000 05480010 05493000 05499000 05529215 05529260
05529280 05529400 05529430 05529460 05529490 05529590 05529630 05529820 05535080 05535180 05535220
05535250 05535440 05535595 05538000 05549000 05551000 05553000 05571000 05598000

SCAN
0078 *02245600*

02248900 02249200 02262700 02263300 02267500 02268400 02275160 02288500

SCANENTER
0002 *00155110*

*02804000**02862400* 04068000

SCANINC
0080 *02333100*

02333500 02333520 02364380 02364382 02364385 02364420 02364460 02364500 02364580 02364620 02371004
02372050 02382300 02404300 02414290 02419250

SCANX
0081 *02495000*

02510000 02621100 02732000 02770000 02776000

SCN
  0002 *00174000*

*02184000**02507000**02508000**02509000**02585000* 02585000 02598000 *02599000**02601000* 02622000 *02676000*
*02723000**02746000**02764000* 02764000 *02764000* 02771000 *02775000* 02781000 *02781000* 02781000 *02781000*
 02806000 *02819000**02821000**02822000**02825000**02827000**02828000**02831000**02837000**02838000**02839000*
*02840000**02841000**02849000**02851000* 02915000 *02916000* 03563920 03563940 03563950 *03570000**04765000*
04770000 04771000 *05067390**05529260**05529400*

SCRAM
  0132 *04488000*

04498100 04563000 04576000

SEARCH
  0006 *00399000*

 00736000 00738000 00739100 01195000 01588000 *01735000**01741000* 01742000 01790000 01984000 02568000
 02862400 03584000 03590000 04678000 04773050 04870000 04982000

SEARCH
  0043 *01195000*

 01205000 01229000

SEENADOUB
  0002 *00169100*

 00691800 00693500 00693600 01085500

SEG
  0006 *01025000*

 01041000 01050000

SEG
  0057 *01525000*

01537000 01542000 01543000

SEG
  0117 *03971000*

03986000 03987000 03988000

SEGDICT
  0090 *02980000*

```
        03314000  03321000  03326000  03327000  03351000

SEGEND
  0004 *00334000*

        01005000  01050000

SEGLS
  0090 *02993000*



SEGMENT
  0006 *00421000*

        00629000 *00937000* 01022000  01699000  03289000  05215000

SEGMENTSTART
  0006 *00778000*

        00795000  01700000  02087000  03128000  05128000  05151000  05532000

SEGMNT
  0090 *02988000*

       *03313000* 03314000  03315000  03317000  03321000 *03325000* 03326000  03327000

SEGNO
  0002 *00245000*

        00755000  00914000  00932000  00975000  00995500  01537000  01589010  01589020  01799000  01801000  03217000
        03252000  03939000  03986000  04433000 *04434000* 04444000

SEGOVF
  0006 *00395000*

        00396000  00930000  01269000  01295000  01355000  01361000  01371000  01388000  01393000  01395000  01397000
        01399000  01411000  01423000  01431000  01527000  01563000  01566000  01575000  01578000 *01653000* 01703000
        01787000  02278300  03990315  03990542  04415040  04518200  04613000  04866100  04920200  05345000  05510000

SEGOVFLAG
  0002 *00169070*

        00783000  01018025  01021957  01661000  01702000

SEGPTOG
  0002 *00169078*
```

01018375 01019000 02270200 02286100 03571300

SEGSTRT
0004 *00333000*

00781000

SEGSW
0002 *00169085*

00794400 01021350 02275600 02432100 03363200 03385350 03385400

SEGSWFIXED
0002 *00169088*

02275600 02431000 03385400

SEGSZC
0002 *00355000*

03240000 03276000 03305000

SEGSZF
0002 *00354000*

03310000 03331000

SEGTYP
0006 *00940000*

00949000 01018370 01019000 01019100

SEGUS
0091 *02991000*

03244000 03251000

SEGO
0090 *02981000*

03291000 03299000 03303000 03338000 03339000 03339100 03348000 03350000 03351000 03352000 03353000
03354000 03355000 03363150 03363700 03364000

SEMI
0002 *00303000*

00625000 01948000 02193000 02199000 02240000 02241000 02827000 02828000 02949000 04493000 04562000
04566000 04571000 04770000 04813000 04909000 04916100 04917000 05028010 05038000 05051100 05097000

05197000 05300290 05300300 05460000 05464000 05471000 05493000 05500000 05529635 05529820 05538000
05595000 05598000

SEMIC
0134 *04657000*

04718100 04765000 04766500 *04770500*

SENSF
0002 *00208135*

02230220

SENSPDEUNF
0002 *00208130*

03162005

SEQ
0006 *00706010*

00706075 00706085 00706120 00706150

SEQBASE
0002 *00170000*

02283400 02283700 02364830 02364840 *02364840* 02374053 02389400 02409400 *02420057* 02420057 02423000
02424000 02424000

SEQCHK
0006 *02321000*

02330000 02331000 02364800 02373045 02374500 02387000 02396100 02407000 02411000 02414380 02416100
02416200 02417000 02419000 02420020

SEQERR
0080 *02362300*

02362850 02420040

SEQERRCT
0002 *00147000*

02420015 *02420015**02420080* 02420080 05617000 05618000

SEQERRORS
0002 *00169065*

00422275 00604000 00611000 02054010 02420000 02420090 02426010

SEQINCR
0002 *00170000*

02283400 02284000 *02284600* 02284600 02364840 02420057 02424000

SEQNUM
0078 *02254600*

02259100 02259700 02281300 02283700 02284000

SEQTOG
0002 *00169064*

02272600 02283200 02364820 02374053 02389400 02409400 02420053 02421000

SET
0078 *02261800*

02267500 02272900

SETLINK
0006 *05063600*

05063730 05089050 05272050

SETPNT
0006 *00723140*

00723325 00723670 00916300

SETUPSTACK
0006 *00916900*

00926000 00965000 00973000 00982000

SGLCC
0090 *03005000*

03328000

SGLCF
0090 *03004000*

SGN

```
  0070 *01819000*
       *01829000* 01852000 *01926000* 01931000 *01935000* 01938000

SGNO
  0006 *01055000*
        01063000  01065000

SGNOC
  0002 *00351000*
        01008000  01041000  01063000  03239000  03275000  03304000

SGNOF
  0002 *00350000*
        03313000  03325000

SGTYPC
  0090 *03000000*


SGTYPF
  0090 *02999000*


SHOW
  0061 *01626100*
       *01631100* 01640000

SHX
  0002 *00225000*
        00226000  00790000  01670000  01737000  01754000  01761000  03990128

SINGLFTOG
  0002 *00169069*
        00422110  00422190  02269600  02274700  02287300  03384000  03791000  05066005

SINGLFTOG
  0114 *03791000*
       *03799500* 03834000
```

SINGLFTOG
0145 *05066005*

05067275 05069000

SINX
0090 *03034000*

SIZ
0070 *01817000*

01844100 01851400 *01851400* 01857100 *01857100* 01869000 01880000 01885000 01891000 01898000 *01898000*
01913050 01913050 01930000 *01942100*

SIZE
0002 *00256000*

00700000 00721000 00772000 01083000 *01084000* 01097000 01162000 01634000 01639000 01646000 01708000
03210000 04415200 04415210 04505200 04506000 05079900 05080000 05455000 05458600

SIZF
0006 *02434000*

02444000

SIZE
0090 *03067000*

03071000 03075000 03082000

SKP
0006 *01262000*

01262000 01264000 01364010 01599000 01625000

SKP
0048 *01255000*

01255000 01257000

SKP
0086 *02638000*

02639000 02642000

SKP
  0087 *02683000*

       02684000 02688000 02689000

SL
  0134 *04655100*

       04681140 *04744100* 04753500

SLASH
  0002 *00302000*

       01856000 01944000 01948000 02196000 02952000 04090000 04133000 04562000 04577000 04649500 04883000
       04883650 04912200 04930000 05097000 05172000 05174000 05179000 05434000 05443000 05460000

SLASH
  0134 *04649500*

       04687560

SLCNT
  0134 *04655500*

       04687520 *04687530* 04687530 04687560 04687600 *04687650**04744000*

SLOWV
  0002 *00208110*

       02230115 02230120 02230140

SND
  0002 *00325000*

       00764280 00923000 03990480 04028040 04311000 04318000 04319010 04558000

SNGL
  0118 *03990040*

      03990480 03990520

SP
  0150 *05220000*

      05223000 05224000 05227000

SPCLBIN2DEC
  0006 *01261100*

```
          01261140  01261150  04604030

SPDEUN
  0090 *02988910*

          03162005  03203000

SPDFUN
  0090 *03042000*

          03043000  03051200

SPDF
  0002 *00208140*

         *02230130**02230140*

SPEC
  0081 *02451000*

          02465000  02473000

SPECCHAR
  0121 *04029000*
          04109000  04114400  04172000 *04188000*

SPIN
  0077 *02149000*

         *02224000* 02239000

SPLINK
  0002 *00155000*

          00764040 *00786000* 00949100  00950000  00959000  00972000  00975000  01007100  01018135  04500100  04500310
          04503000  04773030  04816110  04946000  04964100  05128000 *05132000* 05151000 *05153000* 05212005  05214000
          05221000  05222000  05223000  05291000  05432000  05499100  05503000  05529640  05529660  05529680  05529690
          05529720  05532000 *05534000* 05535230  05535240 *05579000* 05579000  05612100

SPLIT
  0006 *03621000*

          03631000  03743000  04157000

SPP
  0070 *01824000*

         *01875000* 01913500
```

SRI7
  0077 *02172000*

        02173000 02175000 02195000 02200000

SS
  0113 *03640000*

        03664100 03665000 03689000

SSF
  0002 *00330000*

        03132000

SSN
  0002 *00326000*

        01384000 04404500 04415021 04848000 04856000 04863000 04912200 05318400 05347000 05535510

SSNM
  0002 *00169200*

        00169210 00169220 00598040 00598180 00599500 00691510 00691520 00691530 00691540 00691550 00692010
        00692020 00692030 00692040 00706260 00706380 00794400 02263900 02279300 02364360 02364560 02364800
        02380125 02380175 02380200 02380300 02401125 02401175 02401200 02401300 02419000 02420011 02420020
        02420040 02420050 02420060 02420120 02426030 02432300 03374100 03374200 05619050

SSP
  0002 *00326000*

        03990230

STACK
  0121 *04021000*

        04066000 04067000 *04382000* 04392000

STACKHEAD
  0002 *00226000*

        00790000 01671000 01737000 01754000 01761000 01762000 03990128

STAR
  0002 *00302000*

        01896000 02946000 04493100 04883200 04918000 05067210 05116000 05117100 05529270

STAR
  0006 *00723140*

        00723150 00723265 00723295

STATEMENT
  0006 *05136000*

        05366000 *05536000* 05603000 05612000

STD
  0002 *00326000*

        00721000 00812400 00830000 00846000 00848000 00856000 00856300 00877000 00925040 00981000 01454000
        01460000 01465000 01470000 01475000 01490000 01495000 01500000 01504000 01509000 01514000 01732000
        03134000 03139000 03665400 03990545 04028030 04028040 04235000 04245300 04245400 04260010 04282010
        04289005 04292010 04311000 04318000 04329010 04404625 04410000 04415220 04415230 04415240 04415260
        04509400 04510000 04516350 04518000 04520000 04545000 04925000 04979000 05036000 05041200 05146000
        05490000 05506000 05517000

STMTFUN
  0006 *04959000*

        05007000 05017050

STMTFUNID
  0002 *00278000*

        04113000 04965100 04999000

STMTFUNREF
  0006 *03969000*

        03990000 04114100

STN
  0002 *00327000*

        01459000 01499000 01508000 03665400 05057000

STOP
  0006 *05485000*

        05494000 05571000

STOPC
  0090 *03029000*

02716000 02718000 *02718000* 02719000 03879000 03890000 04177500 04177575

STRINGTYPE
  0002 *00296000*

01832000 01844000 01851000 02645000 02675000 02722000 03875000 04176000

STRINGWORD
  0087 *02683000*

02697000 02699000 02706000 02714000

STRT
  0080 *02363000*

02364000 02364910 02367000 02373200 *02375000* 02385000 02388000 02396200 02408000 02408500 02414382
02416300 02419050 02419300

STRTA
  0080 *02362950*

02375600

STRTSFG
  0002 *00182000*

00962000 *00963000* 03265000 03285000

STYPC
  0002 *00340000*

01040000 03237000 03273000 03302000

STYPF
  0002 *00339000*

SUB
  0002 *00326000*

03661000 03734000 03766000 03990310 04277000 04282000 04289000 04289007 04290000 04291000 04292000
04404450 05516000 05522000

SUBC
  0006 *00723140*

00723150 00723230 00942000

SUBCLASS
  0002 *00247000*

00711000 00720000 00723680 00768000 00773100 00854100 00856100 00894000 00895000 00897000 00916320
01018160 01084000 01085500 01135000 01165000 01314000 01323000 01335000 01443000 01988000 *03564000*
03565000 03601000 *03601000* 03606000 *03606000* 03703000 03747000 03772000 03804000 03830000 03834000
03937000 03940000 03976000 03990122 04008000 04075000 04123000 04158000 04536000 04983000 04993000
05069400 05107050 05131000 05227000 05267100 05506000 05529680 05529690 05529700 05529760 05529780
05529790

SUBREF
  0006 *04405000*

04428000 05159000

SUBRID
  0002 *00285000*

00861000 00862000 01018200 03214000 03614000 03616000 03894050 03894100 03912000 03915000 04417000
05533000

SUBROUTINE
  0006 *05530000*

05535000 05572000

SUBSCRIPTS
  0006 *03632000*

03785100 03787000 04128000 04513100 05019000

SUBSVAR
  0002 *00288000*

03878000 03902000 03903000 03930000 03935000 04130000 04133000

SUCH
  0139 *04784000*

04840000 04843000 *04850000*

SUCHTOG
  0139 *04785000*

04850000 04907000 04913000

SUM

0113 *03636000*

*03674000**03691000* 03691000 *03699000* 03699000 03710000 *03710000* 03714000 *03714000* 03721000 03733000
03734000 03737000 03740000 03760000 03765000 03766000 03768000

SUPERMAXCOM
0002 *00214100*

00646200 00646600 *03571100* 05185000 05238000

SWARYCT
0002 *00147500*

00691500 00694150 *01095500*

SYL
0043 *01193000*

*01208000**01219000**01227000* 01233000

SYMBOL
0002 *00198000*

02152000 02154000 02512000 02568000 02570000 *02576000* 02577000 02582000 02583000 02593000 *02596000*
02597000 *02721000**02819000**02820000**02822000**02825000**02827000* 02827100 *02828000**02830000**02831000*
02832000 *02837000**02838000**02839000**02840000**02841000**02849000**02850000* 02858000 02862000 02862400
02914000 02925100 02944000 02965000 02972000 02974000

SZ
0006 *01025000*

01034000 *01034000* 01035000 01039000 01044000 01046000 01051000

SZ
0006 *01625000*

01633000 01637000 01638000 01648000 01649000

SZ
0006 *00939000*

00998000 01001000 01007000 01011000 01017000 01021000

SZ
0037 *01030000*

01030000 01032000 01054000 01055000

SZ

0086 *02633000*

02633000 02636000

SZ
0086 *02638000*

02639000 02642000

SZ
0087 *02683000*

02684000 02690000

SZ
0090 *03062000*

03062000 03064000

T
0002 *00151000*

00650000 00706015 00725000 *00846000* 00848000 *00856000* 00856300 *00895000* 00896000 00898000 *00910000*
00910000 00915140 00945000 01028000 01071250 01176000 01194000 *01280000* 01283000 *01311000* 01317000
01319000 01320000 *01325000* 01327000 01585000 01655000 01736000 01744000 01779000 01963000 01995000
02021000 02159000 *02186000**02225000**02226000**02227000* 02229000 *02230030* 02230050 02230130 02230140
02230220 02230300 02237000 *02237000* 02237000 02436000 02512000 03069000 03090000 *03106125* 03106150
*03123000**03161000* 03162030 03162060 *03288000**03689000* 03691000 *03702000* 03707000 03712000 03793000
03880000 03881000 *03897000* 03898000 *03937000* 03938000 03941000 *03963000* 03963000 03964000 03964500
03994000 04021000 *04415150**04415200* 04415200 04415210 04431000 04487000 *04585000* 04591000 04655000
04773020 04786000 *04982000* 04983000 04984000 *04984000**05023000* 05107050 *05107050**05183000* 05190000
05191000 *05191000**05196000* 05196000 05224000 *05224000**05297000* 05297300 *05446000* 05449000 05456000
05457000 05497000

T
0006 *00647000*

00648000 00649000 00651000

T
0006 *00706015*

00706065 00706090

T
0006 *00401000*

T

0006 *01176000*

01180000 01181000 01182000 01195000

T

0006 *01262000*

01264000 01267000 01273000 01303000 01353000 01359000 01364010 01365000 01378000 01404000 01428000
01437000 01520000 01558000 01570000 01582000 01599000 01612000 01624000 01625000 01704000 01735000
01743000 01758000 01768000 01778000

T

0006 *02321000*

02323000

T

0006 *02436000*

02442000 02443000 02444000

T

0006 *03573000*

03581000 03584000 03590000 03592000 03593000 03600000 03602000 03607000

T

0024 *00653000*

00658000 00661000 00662000 00674000 *00674000*

T

0023 *00650000*

00653000 *00677000* 00680000 00685000 00687000 *00688000* 00688000 *00691000* 00691900 *00691900* 00691900
00691950 *00691950* 00691950 00692000 *00700000* 00701000 00702100

T

0034 *00915140*

00915240 00915280 *00915420* 00915440 00915460 *00915560**00915580* 00915580 00915600 00915660 *00915680*
00915700 00915700 00915720 00915760 00915780 00915800 *00915960* 00916020 00916040 00916080

T

0030 *00725000*

00744000 00744000 00746000 *00746000* 00748000 *00763000* 00763000 00765000 00775000 *00775000*

T

```
  0037 *01028000*
       *01035000* 01050000

T
  0036 *00944000*
        00944000  01024000  01025000

T
  0036 *00945000*
       *00972000**00989000* 00991000 *00991000**01001000* 01005000

T
  0037 *01030000*
        01032000  01054000  01055000

T
  0046 *01237000*
        01239000

T
  0040 *01071250*
        01082000  01085000  01095000  01112000  01114000  01122000  01122500  01122600

T
  0043 *01194000*
       *01217000* 01221000  01222000

T
  0058 *01585000*
       *01588000**01588100* 01588200  01589000  01589010  01590000

T
  0065 *01744000*
       *01745000* 01746000  01747000  01748000 *01748000* 01749000

T
  0069 *01779000*
       *01790000* 01791000  01795000  01800000  01802000 *01811000*
```

T
0062 *01655000*

01656000 01657000 *01662500* 01662700 *01663000* 01664000 *01666000* 01667000 *01669000**01671000* 01672000
01673000 01677000 01680000 01687000 *01689000* 01693000 *01693000* 01695000

T
0064 *01736000*

01737000 01738000 01739000 01740000 *01740000* 01740000 01741000

T
0074 *02021000*

02024000

T
0071 *01963000*

01973000 01974000 01975000 *01975000* 01975000 01976000 01977000 01977100 *01985000* 01985000

T
0072 *01995000*

*02001000**02004000* 02005000 02006000 02007000 *02007000* 02008000 02011000

T
0077 *02159000*

02163000

T
0082 *02489200*

02489200 02489600 02496000 02497000

T
0081 *02512000*

02568000 02570000 02576000 02577000 02582000 02583000 02593000 02596000 02597000 02607000 02631000
02730000 02819000 02820000 02822000 02825000 02827000 02827100 02828000 02830000 02831000 02832000
02837000 02838000 02839000 02840000 02841000 02849000 02850000 02858000 02862000 02862400 02914000
02925100 02944000 02965000 02972000 02974000

T
0080 *02343000*

02346000

T
0080 *02362010*
02362030 02362050 02362120 02362150 02362190

T
0085 *02615000*
02615000 02618000

T
0086 *02631000*
02646000 02655000 02657000 02663000 02666000 *02666000* 02671000 *02671000*

T
0085 *02608000*
02608000 02610000

T
0085 *02607000*
*02620000**02620200* 02620200 02621000 02622100 *02622100* 02622200 02623100

T
0088 *02730000*
*02731000**02733000* 02734000 02735000 02736000 02738000 02739000 02745000 02763000 *02763000* 02778100
02780000 02788000

T
0092 *03069000*
03072000 03073000

T
0090 *03062000*
03064000 03066000 03067000

T
0090 *03090000*
03094000 04051000

T

```
  0092 *03070000*
        03070000

T
  0114 *03793000*
       *03802100* 03803000  03804000  03805000 *03805000**03810010* 03815000

T
  0120 *03994000*
       *04011000* 04015000

T
  0121 *04051000*
        04058000

T
  0121 *04021000*
       *04082000* 04083000  04085000  04089000  04097000 *04139000* 04140000  04142000  04146000 *04224000**04225000*
       *04226000**04227000**04228000**04229000* 04245000  04245300  04245400  04245500 *04245500* 04249000

T
  0136 *04663000*
        04663000  04664000

T
  0134 *04655000*
        04681000  04681002  04681004  04681010 *04681010* 04681030  04681060  04681120  04681140  04684000  04687530
        04688000  04695500  04707000  04708500  04708520  04708525  04730120  04730125  04730130  04730137  04744500

T
  0131 *04431000*
       *04440000* 04442000  04448000 *04462000* 04465000

T
  0138 *04773020*
       *04773050**04773060* 04773080  04773140  04773150 *04773150**04773160* 04773160

T
```

```
  0132 *04487000*
       *04500600* 04501000 *04503100* 04503200  04505200  04506000  04509000 *04509150* 04509200  04509250 *04531000*
        04532000

T
  0139 *04786000*
       *04885000**04886100* 04890000  04899000

T
  0134 *04657500*
        04658000

T
  0156 *05497000*
       *05503000* 05506000

TA
  0078 *02254900*
        02256100  02257600  02257900  02259400

TA
  0078 *02249800*
        02250500  02251900  02252900  02253100

TA
  0080 *02333160*
        02333400  02333420  02333460

TAPE
  0006 *00389000*
        00391000  02367000  02383010  02386000  02396000  02402000  02416000  03366100

TAPETOG
  0002 *00168400*
       *02265700**02272510**02272560* 02365000 *02373030* 02383010  02385000

TB
  0002 *00184000*
        02254900  02367000  02386000  02387000  02395000  02396000  02402000  02407000  02411000  02415000  02416000
```

02416200 02417000 02491700 02492200 02492600 04668000 04668600 04670500

TB
0078 *02254900*

02257300 02257900 02259100 02259400

TEMPNEXT
0159 *05537100*

05540000 05548000 05552000 05560000 05561000 05568000 05578100

TEN
0002 *00210000*

02754000 02758000 02791000 02792000 02796000 02797000 02870000 02874000 03108000 03438000

TEST
0002 *00228000*

01976000 01978000 01980000 04598000 *04868000* 04870000 04871000 04873000 *05584000* 05590000 *05590000*
05591000

TEST
0133 *04598000*

04601000 04604010 04604025 *04604030* 04604040 04604050

THEBIGGEST
0090 *02983000*

03152000 *03152000* 03260000

THRU
0002 *00316000*

00422190 00422200 02571000 02591000

TIMETOG
0002 *00169066*

02276800 05614000

TIPE
0002 *00176500*

01019160 01019170 01985000 02862000 02939100 03433100 05529475 05529570 05529680 05529760

TIS
  0002 *00359000*

       00376000 00380000 00389000

TIX
  0141 *04964000*

       04993510 04993570 *04993595*

TM
  0121 *04028010*

       04114520 04115000 *04260000* 04260010 *04261000**04268000**04270000**04271000**04281000**04282000* 04282010
       04289000 04289005 04289007 *04291000* 04291010 *04292000* 04292010 *04310000**04319000* 04319010 04319020
       *04320000**04338000**04339000*

TMPNXT
  0006 *05479000*

       05480020

TN
  0034 *00915140*

       00915220 00915280 *00915440* 00915480 *00915480**00915560**00915580* 00915600 00915620 *00915680**00915700*
       00915720 00915760 00915780 00915800 *00915960* 00916000 00916020 00916080 *00916160* 00916300 00916320
       00916340

TOADDR
  0002 *00268000*

       00695000 01108000 01795000 01811000 02187000 04435000 04527000 04587000 04773060 04773140 04999000
       05048000 05091000 05093000 05186200 05238300 05274000 05276000

TOADINFO
  0002 *00273000*

       00744000 02188000 03836000 03954000 03990108 04009000 04475000 05196000

TOADJ
  0002 *00265000*

       00672000 05535240

TOBASE
  0002 *00270000*

       00670000 01610000 01680000 03363700 03607100 04876000 04893000 04902000 05000000 05228000

TOCE
0002 *00261000*

00672000 02188100 04604040 05085000 05093000 05186200

TOCLASS
0002 *00263000*

00695000 01134000 01588100 01588400 01811000 01812100 01977100 02187000 02188100 03386000 03586000
03605000 03893000 03900000 03909000 03915000 03990116 03990134 04015000 04480000 04493200 04588000
04604040 04678500 04679400 04773060 04773175 04800100 04871000 04873000 04883350 04900000 04965100
04986000 04987100 04999000 05017020 05025200 05029400 05107100 05133000 05141500 05168100 05177100
05186200 05194000 05238300 05285000 05391200 05397200 05403200 05437000 05442500 05521200

TOCNTRL
0134 *04652500*

04750000

TOCODF
0134 *04650000*

04687560 04690000 04719500 04726530 04732000 04747400 04758500

TODECIMAL
0134 *04652000*

04740000 04759000

TOEQ
0002 *00263100*

00672000 05268000 05270500 05276000

TOFORMAL
0002 *00266000*

04446000

TOG
0077 *02140000*

*02207000**02209000**02211000**02212000**02213000**02214100**02214500**02215000* 02216000 *02217250**02217275*
02217350 02217450 *02225000**02226000**02227000* 02228000

TOG
0113 *03638000*

03675000 03692000 03696000 *03697000* 03705000 03712000 *03717000* 03717000 03722000 03731000 03760000

```
          03763000

TOLASTC
  0002 *00262100*

          05085000  05268000

TOLINK
  0002 *00269000*

          04435000  04651500  05085000  05268000

TOLINK
  0134 *04651500*

          04739500

TONEXTRA
  0002 *00272000*

          03836000  03953000  03990108  04009000  04474000  05000000  05017030

TONUM
  0134 *04653000*

          04749800

TOO
  0084 *02540000*

          02546000

TOP
  0002 *00330010*

          03990555

TORFLADD
  0002 *00274000*

          05268000

TOREPEAT
  0134 *04650500*

          04687560  04692000  04719500  04726500  04726530  04732500  04747600  04759000

TOS
```

0002 *00360000*

00376000 00379000 00388000

TOSEGNO
0002 *00262000*

00914000 00956000 00966000 00980000 00984000 03987000 03988000 04415090 04446000 04527000 04549000
04773070 04773140 04816136 04923000 04998000 05041400 05041700 05048000

TOSIGN
0134 *04653500*

04732500

TOSIZF
0002 *00271000*

00696100 00702100 01136000 03106150 05079400 05080000

TOSUBCL
0002 *00264000*

03893000 03901000 03990117 04015000 04987000 04987100 05069200 05080500 05091000 05132000 05134000
05227000 05274000 05529240 05529340

TOTAL
0134 *04655500*

04680500 04684000 *04685500* 04685500 04686000 04687560 04687590 *04687600* 04687600 04689500 04690000
04693000 *04693000* 04695500 04697500 *04697500* 04698000 04719500 04726000 04726530 04726540 *04726540*
04732000 04739500 04740110 04747400 04748800 *04749400* 04749400 *04749800* 04749800 04758500 04766000
04766000 04766500 *04766500* 04772600 04772615

TOTYPF
0002 *00267000*

04986000 04998000 05080500 05132000 05133000 05227000

TOWIDTH
0134 *04651000*

04726000 04732500 04747400 04749400 04758500

TP
0006 *00391000*

02367000 02383010 02386000 02396000 02402000 02416000

```
TPHASE
  0134 *04648000*

         04746050  04762500

TRACEBACK
  0006 *01599000*

         01611000  01639000  01647000  01657000

TRANSFER
  0006 *00706525*

         00706800  00915420

TRB
  0002 *00328000*

         00764240  01364035  04516300

TS
  0002 *00151000*

         01779000  02140000  02730000  02866000  02867000  02869000  04773020

TS
  0069 *01779000*

        *01791000* 01793000  01799000 *01800000**01801000* 01803000

TS
  0077 *02140000*

        *02217050* 02232000

TS
  0088 *02730000*

         02752000  02761000  02783000

TS
  0138 *04773020*

        *04773110* 04773120  04773130 *04773130* 04773130

TSEGSZ
  0002 *00337000*

         01021000 *01021000**01051000* 01051000  02988000  05619000
```

```
TSFGSZ
  0090 *02988000*

       *03331000* 03331000  03358000

TSSED
  0006 *00422270*

        00422285  02656100  02708100  03894060  04406030  04768500  05468100

TSSEDIT
  0006 *00422255*

        00422268  00422280

TSSEDITOG
  0002 *00169082*

        02269300  02277400  02374055  02380000  02401000  02414175  02426000  02431100  02491600  02492450  02492550
        02656100  02708100  03894050  04406030  04667500  04669500  04670100  04768000  05468100

TSSEDITS
  0080 *02342010*

        02342040  02374060  02426000

TSSMES
  0006 *00422235*

        00422240  00422250  00598050

TSSMESA
  0002 *00242220*

        00598050

TSSMESTOG
  0002 *00169086*

        00598050  02269300  02277100

TT
  0034 *00915140*

        00915240  00915260  00915300  00915320  00915600 *00915600* 00915720 *00915720**00915760* 00915760  00915780
       *00915780* 00915800 *00915800* 00916020 *00916020**00916080* 00916080
```

TV
0002 *00199100*

00691520 *00692020* 01071500 *01078000* 01080000 01093000 01119000 01122550 *01122600**04493200**04883350*
04956040 04956100 04956200

TWOD
0002 *00250000*

01162000 01278000 01309000 03719000 04136000 04155000

TWODBIT
0024 *00653000*

00656000 00672000

TWODPRT
0006 *00398000*

04138000

TWODPRTX
0002 *00148000*

00398000 03114000 03123000 *04138000* 04138000

TYP
0006 *05479000*

05480010

TYP
0070 *01822000*

01829000 01830000 01856000 01896000 01914000 01926000 01927000 01933000 01935000 01936000 01940000
01943000 01944000 01948000 01949000

TYPC
0070 *01823000*

01838000 01844000 01864000 01869000 01889000 01921000 01930000 01954000

TYPE
0002 *00162000*

01029000 02029000 02032000 *03389000* 04963000 05067210 05067280 *05067320* 05067320 05067330 05067370
05067380 05069100 05069200 05069400 05073000 05075000 05078000 05080500 05083000 05107030 05107050
05131000 05131000 05132000 05134000 *05164000* 05354000 *05480010* 05529660 *05551000**05558000*

TYPE
0037 *01029000*

01034000 01040000

TYPE
0141 *04963000*

*04983000**04985000* 04987000 04987100 04993000 *04993000* 04993510 04993530 04993560

TYPF
0154 *05354000*

05359000 05361000 05371000

TYPEFIXED
0002 *00252000*

00768000 *01110000**01140000**01144000**01147000**01153000* 05076000 05107050

TYPES
0002 *00234000*

00723670 00768000 00916300 03392000

TYPIT
0006 *05479000*

05480040 05481000 05482000 05483000 05484000 05484500 05548000 05552000 05560000 05561000 05568000

T1
0076 *02064000*

02070000 02072000

T1
0084 *02524000*

02525000

T1
0090 *02988000*

02993000 *03106125* 03106150 *03106150**03106200**03107000**03158000**03184054**03185000* 03203000 *03213000*
03214000 *03230000* 03232000 *03243000* 03244000 *03247000**03279000**03299000**03310000* 03312000 03313000
03315000 03318000 03325000 03328000 03329000 03329100 03329200 03330000 03331000 *03336000* 03337000

T1
  0121 *04031000*

        04209000 04214000 04219000 04257000 04278000 04307000 04326000 04347000 04367000

T1
  0152 *05300170*

       *05300190**05300250* 05300250

T2
  0084 *02524000*

        02528000

T2
  0086 *02631000*

       02652000 02655000 02663000 02665000 02666000 02669000 02671000 02672000

T2
  0121 *04031000*

        04209000 04214000 04219000 04223000 04257000 04278000 04294000 04295000 04296000 04297000 04307000
        04326000 04344000 04347000 04367000

U
  0072 *01995000*

        02006000 *02006000*

UNKNOWN
  0002 *00275000*

        00737000 00739000 00739200 00833000 01109000 01990000 03585000 03598000 03610000 04899000 04997000
        05016000 05071200 05436000

UNPRINTED
  0002 *00169083*

        00422275 02053000 02287000 02289500 02289600 02363100 02380240 02389600 02401240 02409600

UNSEEN
  0139 *04787000*

        04790000

UPARROW

0002 *00303000*

02945000

V
0006 *00646100*

00646200

V
0006 *00706700*

00706720

V
0006 *00646500*

00646700

VAL
0078 *02261500*

*02267500**02268400* 02269300 02269600 02269900 02270200 02270500 *02270800* 02272300 02272510 02272560
02273800 02274100 02274700 02275100 02275300 02275600 02275900 02276500 02276800 02277100 02277400
02277700 02278000 02278600 02279200 02279300 02280100 02280300 02281000 02282900 02283200 02283300
02285200 02285500 02285800 02286100 02286400

VALREQ
0006 *04020000*

04078100 *04078100* 04084000 *04128000* 04128000 04131000 04177000 *04184200* 04184200

VARF
0113 *03638000*

03651000 03656000 03665100 03665400 03675000 03676000 03702000

VARF
0114 *03791000*

03806000 03812000 03827000 03837000

VARIABLEDIMS
0006 *00707000*

00723000 00910000

VARID
0002 *00277000*

00833000 00851000 00854100 00856100 00871000 00880000 01109000 01134000 01165000 01277000 01307000
01446000 01988000 03803000 03880000 03920000 03930600 03931000 03936200 03943200 04085000 04089000
04090000 04498000 04500600 04986000 04987100 05025200 05029400 05071300 05107100 05133000 05446000

VLU
0006 *00636000*

00638000

VLU
0078 *02254600*

02259400

VLU
0090 *03084000*

03084000 03086000 03087000

VOIDSFQ
0002 *00169220*

02263900 02279300 02419000

VOIDSFQ
0078 *02249500*

02250100

VOIDTOG
0002 *00169058*

02263900 02279?00 02419000 02419050

VOIDTSEQ
0002 *00161010*

02264800 02280300 02373045 02396100 02416100

VOIDTTOG
0002 *00169092*

02264800 02280100 02373045 02373100 02373200 02396100 02396200 02416100 02416200

VPHRASE
0134 *04642510*

04710500 04746750 04757400 04760550

VRB
  0134 *04656100*
       04726520 04732100 *04735740**04746750* 04747700 04759150 *04760890*

VT
  0056 *01439000*
       01443000 01444000 01450000 01452000 01454000 01469000 01470000 01471000 01481000 01483000 01490000

W
  0006 *00415000*

W
  0006 *01758000*
        01761000 01763000

W
  0006 *01768000*
        01771000 01773000

WARN
  0079 *02262100*
        02287300 02287600

WARNCOUNT
  0002 *00242750*
        00422280 *00422280* 05618100 05618110

WARNED
  0002 *00169087*
        02431100 05618100

WB2D
  0046 *01237000*
        01247000 01249000

WIDTH

0134 *04655500*

04710000 *04710500**04729500**04735730**04746750* 04757000 04757420 *04757420* 04758410 *04758410* 04758500
04760980

WOP
0002 *00183000*

01200000 01202000 01204000 03419000

WRAP
0139 *04784000*

04911000 04917100 *04950000*

WRAP
0155 *05431000*

WRAPUP
0006 *02976000*

03367000 05613000

WRITALIST
0006 *00422120*

00422210 00600050 00600080 00622000 00735000 00755000 00767000 00781000 01005000 01050000 01065000
01232000 01434000 02024000 02972000 03204000 03209000 03244000 03251000

WRITAROW
0006 *00422100*

00422285 00598200 01250000 02053000 02054010 02380240 02389600 02401240 02409600

WRITEDATA
0006 *00418000*

01023000 01052000 01633000 01643000 03105000 03106210 03108000 03111000 03124000 03260000 03890000
04934000

WRITEPGM
0036 *00947000*

00949000 *01000000*

WSA

0134 *04654000*

04680500 04684000 04686000 04687560 04687590 04687600 04690000 04692000 04695500 04698000 04719500
04726530 04732000 04739500 04740100 04740110 04747400 04748800 04749400 04749800 04758500 04772000
04772600 04772620

W1
0006 *00723140*

00723160 00723450 00796000 00927000 00938000 00939000

W2
0006 *00723140*

00723280 00723450 00927000 00938000 00939000

W2
0006 *00706525*

00706530

W2
0034 *00915180*

00915240 00915260 00915300 00915320 00915420 00915600 00915720 00915760 00915780 00915800 00916020
00916080 00916300

W3
0034 *00915180*

00915220 00915240 00915280 00915300 00915480 00915600 00915720 00915760 00915780 00915800 00916020
00916080 00916160

X
0006 *00400010*

X
0007 *00433100*

00433300 00600020 00600040 00641000

X
0006 *00422270*

00422280

X

```
  0006 *00422255*
         00422260  00422270  00423000

X
  0006 *00927000*
         00932000  00935000

X
  0007 *00426000*
         00429000

X
  0006 *01364010*
         01364035 *01364035*

X
  0006 *04404125*
         04404275  04404350  04404435  04404475  04404600

X
  0006 *05603110*
          05603120  05603130

X
  0052 *01407000*
        *01415000* 01426000

X
  0069 *01779000*
        *01800000* 01801000  01802000

X
  0063 *01706100*
        *01708400* 01733100

X
  0071 *01963000*
        *01973000* 01974000  01974500  01984000  01985000  01987100  01988000 *01988000*
```

X
  0112 *03574100*

       03607000 03607100

X
  0120 *03993000*

       04007000 04008000 04011000

X
  0157 *05529120*

        05529140 05529250 05529420 05529620 05529635 *05529820*

XCH
  0002 *00326000*

        01318000 01340000 01343000 01486000 01489000 01496000 01498000 01507000 03751000 03754000 03776000
        03779000 03904100 03928000 03934000 03936400 03943300 03990280 03990340 03990360 03990440 03990480
        03990555 04028030 04028040 04152000 04162000 04165000 04199000 04235000 04240000 04245300 04248000
        04260010 04263000 04282010 04284000 04289005 04291010 04292010 04299000 04300000 04311000 04313000
        04318000 04319020 04319030 04329010 04332000 04339010 04404350 04993550 04993590 05372000

XFFR
  0084 *02524000*

        02530000 02577000 02597000

XFFRA
  0084 *02531000*

        02539000 02587000

XGLOBALS
  0002 *00169098*

        00706300 01018110

XHFDB
  0004 *00333090*

        01018140

XHFDF
  0004 *00333070*

        01018170

```
XHEDG
  0004 *00333080*

       01018150

XHEDM
  0004 *00333050*

       01018210

XHEDS
  0004 *00333060*

       01018200

XIT
  0002 *00327000*

       00676000  00710000  01073000  01127000  01274000  01584000  01613000  01626100  01706010  01736000  01744000
       01785000  01824000  01996000  02062000  02140000  02245900  02249800  02333180  02345000  02363000  02523000
       02607000  02632000  02682000  02728000  02802000  03575000  03635000  03990040  03995000  04021000  04404150
       04406020  04488000  04773020  04784000  04964000  05010000  05063620  05103000  05138100  05214000  05232100
       05300160  05302000  05431000  05508000  05535040  05537000

XIT
  0028 *00710000*

       00711000 *00722000*

XIT
  0023 *00676000*

       00697200 *00705000*

XIT
  0041 *01127000*

       01167000 *01170000*

XIT
  0049 *01274000*

       01285000  01298000 *01301000*

XIT
  0040 *01073000*

       01074000 *01124000*
```

```
XIT
  0058 *01584000*

       01587000  01588300 *01595000*

XIT
  0064 *01736000*

       01739000 *01741000*

XIT
  0061 *01626100*

       01629000 *01651100*

XIT
  0060 *01613000*

       01618000 *01622000*

XIT
  0063 *01706010*

       01708000 *01733105*

XIT
  0069 *01785000*

       01789000  01791100 *01813000*

XIT
  0065 *01744000*

       01747000 *01749000*

XIT
  0072 *01996000*

       02002000 *02012000*

XIT
  0078 *02245900*

       02246900  02247000  02247400  02248600

XIT
```

```
  0076 *02064000*
         02068000  02076000

XIT
  0076 *02077200*
         02077400  02077700

XIT
  0076 *02062000*
         02064000  02077200  02080000  02089000  02093000 *02116000*

XIT
  0070 *01824000*
         01944000 *01950000*

XIT
  0077 *02140000*
         02181000  02185000  02185010  02192000  02194000  02199000 *02240000*

XIT
  0078 *02249800*
         02253100  02254000

XIT
  0080 *02345000*
         02357000  02359000

XIT
  0081 *02802000*
         02827300  02828000  02830000  02843000  02847000  02849000  02860000  02862600  02884000  02899000  02900000
         02901000  02920000  02924100  02926000  02930000  02931000  02939000  02943000  02945000  02946000  02947000
         02948000  02949000  02950000  02952000  02961000  02963000  02964000  02966000 *02971000*

XIT
  0084 *02523000*
         02559000  02560000  02560100  02569000  02573000  02589000  02593000 *02602000*

XIT
  0087 *02682000*
```

```
        02686000  02707500  02711000 *02724000*

XIT
  0088 *02728000*
        02733500  02739000  02743000  02746000  02747000  02775000  02778100  02788000 *02799000*

XIT
  0086 *02632000*
        02655500  02660000 *02677000*

XIT
  0080 *02333180*
        02333260  02333300  02333320  02333500

XIT
  0087 *02686000*
        02693000  02695000  02697000

XIT
  0085 *02607000*
        02621000  02622000  02622100  02623100 *02627000*

XIT
  0080 *02363000*
        02376000 *02430000*

XIT
  0118 *03990040*
        03990090  03990105  03990124  03990136  03990200 *03990810*

XIT
  0112 *03575000*
        03588000  03594000  03603000  03611000 *03618000*

XIT
  0113 *03635000*
        03648000 *03784000*

XIT
```

```
  0120 *03995000*
         03996050  04005000 *04018000*

XIT
  0121 *04021000*
         04078000  04093000  04106000  04121000  04129000  04153000  04186000  04204000  04208100  04212000 *04395000*

XIT
  0129 *04404150*
         04404200  04404250 *04404675*

XIT
  0130 *04406020*
         04415100  04415110  04415130  04415160 *04427010*

XIT
  0132 *04488200*
         04488400  04488500  04581000

XIT
  0138 *04773020*
         04773040  04773090  04773100 *04773180*

XIT
  0139 *04784000*
         04788000  04813000  04836000  04858000  04883100  04883700  04904000  04909000  04915000  04918100  04931000
         04947000 *04957000*

XIT
  0132 *04488000*
         04488200  04493000  04498200  04562000  04571000  04574000 *04578000*

XIT
  0146 *05103000*
         05106000 *05124000*

XIT
  0148 *05138100*
```

```
          05141000  05144000 *05147100*

XIT
  0140 *04788000*
          04796000  04802000 *04811000*

XIT
  0144 *05063620*
         *05063625* 05063680

XIT
  0141 *04964000*
          04969000  04975000 *05005000*

XIT
  0142 *05010000*
          05017050  05027000  05028010  05038000  05051100 *05062000*

XIT
  0153 *05302000*
          05309000  05319000  05326000  05330000  05332000 *05350000*

XIT
  0158 *05535040*
          05535090  05535110  05535225  05535235  05535450  05535570 *05535605*

XIT
  0152 *05300160*
          05300180  05300220  05300240  05300260 *05300300*

XIT
  0159 *05537000*
          05578000 *05602000*

XIT
  0151 *05232100*
          05237000  05243000  05262000  05282000 *05287100*

XIT
```

0155 *05431000*

05435000 05444000 05448000 05451200 *05465000*

XITCODE

0156 *05497000*

*05507000**05508000* 05514000 05518000

XPHASF

0134 *04648500*

04747400 04749000 04749200

XR

0002 *00185100*

00706260 00706280 00706300 00706450 01018160 01018170 01018180 01018190 01018200 03391100

XR

0006 *00706730*

00706735 00706800

XR

0006 *00706010*

00706020 00706075 00706120 00706150

XRBUFF

0002 *00169320*

00169330 00392700 00392800 00706350 00706780 00915360 00915520 01018125

XRBUFFDIV3

0002 *00169320*

00706340 00706780 00915360

XRFF

0002 *00169030*

01018355 01588400 01812100 01977100 01987100 02188100 *02270500**02282900* 03990134 04082100 04114050
04117100 04126100 04415165 04417100 04500655 04535500 04604040 04679400 04773175 04800100 04873000
04965100 04987100 05017055 05025100 05029200 05069300 05071100 05117100 05141500 05144100 05168100
05177100 05243200 05297300 05313200 05391200 05397200 05403200 05442500 05447500 05521200 05535120
05535250 05535540

XRFFCORESORT

0006 *00915100*

00916480 01018340

XRFFF
0006 *00392700*

00706350 00706780 00915420 01018125 01018140 01018160 01018350

XRFFG
0006 *00392800*

00706300 00706735 01018150

XRI
0002 *00185100*

00706340 00706350 00706450 *00706450* 00706750 *00706760* 00915160 00915180 00915360 00915520 00916120
00916460 01018330 *01021957*

XRRY
0002 *00169330*

00706350 00706360 00706380 00706450 00706780 00706800 00915420 00915480 01018125

XRT
0002 *00325000*

01296000 01372000 01564000 01576000 01662500

XTA
0002 *00179000*

00598040 00598160 *00682000**00702000**00705100* 00723610 *00723720**00751000**00834000**00899000* 00916180
*00916400**00949200**01018260**01102000**01445000**01517000**01586000**01620000**01628000**01788000**01971000*
01971010 *01980000**01989000**02083100**02152000**02154000* 02168000 *02178000* 02185000 02185010 02186000
02188100 02203000 02207000 02208000 02209000 02210000 02211000 02212000 02213000 02214000 02214100
02214500 02215000 02217100 02217250 02217275 02217350 02218000 02220000 02221000 02222000 02223000
02224000 02225000 02226000 02227000 02230000 02230040 02230060 02230115 02230120 02230130 02230140
02230200 02231000 02232000 02233000 02234000 *02236000**02236010**02236020**02237021**02237050**02364130*
*02380175**02401175**02414200**02506030**02559000**02593000**02655500**02707500**02739000**02749000**02752000*
02752000 *02775000**02778100**02782000* 02783000 *02783000**02788000**02818030**02827100**02864000**02866000*
02866000 *02869000* 02869000 *02925100**02965000**02974000**03148000* 03150000 *03150000* 03151000 03152000
*03227000**03233000**03262000**03581000**03618100**03648000**03669000**03673000**03821000**03894000* 03894060
*03965000**03978000* 03984000 03990140 *03990165**03990195**04086000**04415120* 04415165 04417100 *04425000*
*04433000**04445000**04445200**04459000**04466000**04467000**04496000* 04500655 04500750 *04537000**04569000*
*04603000**04677500**04680700* 04681004 *04681004* 04708520 *04708520**04708530* 04708530 *04730120* 04730120
04730140 04730140 *04761000* 04768500 *04773050**04795000**04810000**04904000* 04990000 *05017010**05024000*
05070000 05074000 *05130150* 05144100 *05250000**05262300**05266000**05270100**05279000* 05313200 *05439000*
*05467100**05499110* 05535120 05535250 05535540 *05612200*

```
XTA
  0077 *02158000*
       02160000  02163000

Y
  0006 *00927000*
       00932300

Z
  0002 *00151000*
       01407000 *04001000* 04002000 *05168000**05174000**05177000* 05183000  05187000

Z
  0006 *01253000*
       01260000

Z
  0052 *01407000*
       01408100 *01426650*

ZF
  0134 *04656500*
       *04708000**04722500* 04725550 *04727500* 04733500  04735715  04744100 *04744110* 04758000  04762510 *04764000*

ZP1
  0002 *00327000*


 LABEL  OPROGRAMOLISTING00177202CC USER=SITE    ; COMPILE FORTRAN/DISK WITH ALGOL LIBRARY;ALGOL STACK=100 ALGOL  /FORTRAN
```

```
?USER = SITE

        PACKET     2
        INPUT    324 CARDS FROM CRA
        TIME    1103
        DATE   77202 THURSDAY, 07/21/77



*** BURROUGHS B5700 DCMCP MARK XVI.0.73  AND INTRINSICS MARK XVI.0.00 ***

    #NO MESSAGES TODAY

 ?USER= SITE
 ?EXECUTE PATCH/MERGE
 ?DATA CARD
        5:PATCH/MERGE= 1 BOJ 1103 02/06/75
        CDA IN CARD:PATCH/MERGE= 1
        PBD0003 OUT 011 LINE:PATCH/MERGE= 1
        DKA OUT SER MASTERF SITE:PATCH/MERGE= 1
        DKA IN SER PATCH FORTRAN:PATCH/MERGE= 1
        DKA OUT SER CARD001 FORTRAN:PATCH/MERGE= 1
        DKA LOK CARD001 FORTRAN:PATCH/MERGE= 1
        DKA IN SER CARD001 FORTRAN:PATCH/MERGE= 1
        DKA OUT SER CONTROL SITE:PATCH/MERGE= 1
        DKA REL PATCH FORTRAN:PATCH/MERGE= 1
        CDA REL CARD:PATCH/MERGE= 1
?END.
        DKA LOK CARD001 FORTRAN:PATCH/MERGE= 1
  CC USER=SITE;
  COMPILE FORTRAN/DISK WITH ALGOL LIBRARY;
  ALGOL STACK=1000;
  ALGOL FILE TAPE=SYMBOL/FORTRAN DISK SERIAL;
  CORE=12000;
  FILE LINE=LINE PRINT OR BACK UP;
  ALGOL FILE CARD=CARD001/FORTRAN SERIAL;
  ALGOL FILE CARD=CARD001/FORTRAN SERIAL;
  END
        4:ALGOL/FORTRAN= 2 BOJ 1103 03/17/75
        PBD0003 REL 011 LINE 555:PATCH/MERGE= 1
        DKA REL CONTROL SITE:PATCH/MERGE= 1
        DKA REL MASTERF SITE:PATCH/MERGE= 1
        DKA IN SER CARD001 FORTRAN:ALGOL/FORTRAN= 2
        PATCH/MERGE= 1 EOJ 1103
        FOR PATCH/MERGE= 1: PST= 16 IOT= 21 PRO= 14
        DKA IN SER SYMBOL FORTRAN:ALGOL/FORTRAN= 2
        PBD0003 OUT 021 PROGRAM LISTING:ALGOL/FORTRAN= 2
        DKA OUT SER DSK1 SITE:ALGOL/FORTRAN= 2
        DKA OUT SER DSK2 SITE:ALGOL/FORTRAN= 2
        DKA OUT RDM FORTRAN DISK:ALGOL/FORTRAN= 2
        DKA LOK FORTRAN DISK:ALGOL/FORTRAN= 2
        DKA REL CARD001 FORTRAN:ALGOL/FORTRAN= 2
        DKA REL SYMBOL FORTRAN:ALGOL/FORTRAN= 2
```

```
DKA OUT SFR DSRT1 SITE:ALGOL/FORTRAN= 2
DKA OUT SFR DSRT2 SITE:ALGOL/FORTRAN= 2
DKA RFL DSRT1 SITE:ALGOL/FORTRAN= 2
DKA RFL DSRT2 SITE:ALGOL/FORTRAN= 2
DKA OUT SFR DSRT1 SITE:ALGOL/FORTRAN= 2
DKA OUT SFR DSRT2 SITE:ALGOL/FORTRAN= 2
DKA REL DSRT1 SITE:ALGOL/FORTRAN= 2
DKA REL DSRT2 SITE:ALGOL/FORTRAN= 2
DKA REL DSK2 SITE:ALGOL/FORTRAN= 2
DKA REL DSK1 SITE:ALGOL/FORTRAN= 2
PBD0003 REL 021 PROGRAM LISTING 17882:ALGOL/FORTRAN= 2
ALGOL/FORTRAN= 2 EOJ 1115
FOR ALGOL/FORTRAN= 2: PST= 582 IOT= 295 PRO= 126
PKT#0002 REMOVED
```

BURROUGHS B-5700 PATCH/MERGE PROGRAM MARK XVI.0.00   THURSDAY, 07/21/77, 11:03 AM.

********** INPUT **********

```
$@ CARD LISTI CONFLICTS MERGE ZIPARRAY
$, 32 PATCHES FOR FORTRAN.XVI

                                CARD INPUT IS CARD
                                PATCHES/FORTRAN IS NOT ON DISK
                                PATCH  /FORTRAN WILL BE MERGED

$*COMPILE FORTRAN/DISK WITH ALGOL LIBRARY
$*ALGOL STACK = 1000
$*ALGOL FILE TAPE = SYMBOL/FORTRAN DISK SERIAL
$*CORE = 12000
$* FILE LINE = LINE PRINT OR BACK UP
$* DATA CARD
$*   ***** THIS LISTING SHOWS PATCHES IMPLEMENTED AT UCSC *****
$RESET LIST SET TAPE
$SET LISTA SINGLE PRT XREF

$#PATCH NUMBER 101 FOR FORTRAN CONTAINS 1 CARD                                      P  101

        VRB != ASK != LISTEL != TRUE; GO ROUND;                      04735740 P  101
$; THIS PATCH CORRECTS AN FORMAT ERROR IF THE ONLY LIST ITEM                        P  101
$; REQUIRED WAS FOR A VARIABLE FORMAT PHRASE.                                      P  101
$;************************************************************************** P  101


$#PATCH NUMBER 102 FOR FORTRAN CONTAINS 2 CARDS                                     P  102

IF NTAPTOG THEN BEGIN CLOSE(TAPE);%CLOSE TAPE BEFORE NEWTAPE        03366100 P  102
     LOCK(NEWTAPE,*); END;                                          03366200 P  102
$;   THIS PATCH WILL ALLOW THE NEW TAPE FILE TO HAVE THE SAME MFID AND      P  102
$; FID AS THE TAPE FILE.                                                     P  102


$#PATCH NUMBER 103 FOR FORTRAN CONTAINS 7 CARDS.                                    P  103

        IF T≠ EQUL AND T≠ NEQL THEN FLAG(54)                         04245000 P  103
```

```
          ELSE                                                                  04245100 P  103
              BEGIN                                                             04245200 P  103
              EP(9,STD); FO(XCH); EOL(9); EO(T);                                04245300 P  103
              EP(9,STD ); FO(T); EOL(9);                                        04245400 P  103
              T←(IF T=EQUL THEN LND ELSE LOR); CODE←0;                          04245500 P  103
              END;                                                              04245600 P  103
$! BY KFK                                                                              P  103
$! DATE 7/16/74                                                                       P  103
$! TR 2113                                                                           P  103
$!  THIS PATCH WILL ALLOW THE USE OF COMPLEX EXPRESSIONS IN THE                       P  103
$! LOGICAL IF STATEMENT AND IN LOGICAL STATEMENTS IN GENERAL.                        P  103
$! THE  .EQ.  AND .NE.  OPERATORS CAN NOW BE USED TO COMPARE TWO                     P  103
$! LOGICAL ITEMS.                                                                   P  103
$!  SYNTAX EXAMPLES:                                                               P  103
$!  COMPLEX C1,C2                                                                  P  103
$!  LOGICAL L1                                                                     P  103
$!  IF(C1 .EQ. C2) GO TO 1000                                                      P  103
$!  L1 = C1 .NE. C2                                                                P  103


$#PATCH NUMBER 104 FOR FORTRAN CONTAINS 1 CARD.                                        P  104

   IF A.CLASS GEQ 13 THEN FLAG(34) ELSE                                 01130500 P  104
$! BY DDD                                                                           P  104
$! DATE 8/29/74                                                                     P  104
$! THIS PATCH WILL PREVENT THE FORTRAN COMPILER FROM BEING DS*ED                     P  104
$! DUE TO AN INVALID INDEX WHEN THE COMPILING PROGRAM                               P  104
$! REFERS TO A VARIABLE IN AN INCONSISTENT MANNER.  THE COMPILER                    P  104
$!  WILL NOW FLAG THE CONDITION WITH AN ERROR MESSAGE.                             P  104


$# PATCH NUMBER 105 FOR FORTRAN CONTAINS 2 CARDS.                                      P  105

          IF INF.SUBCLASS≥DOUBTYPE AND CL=VARID THEN FMINUS←FMINUS+1;    00854100 P  105
          IF INF.SUBCLASS≥DOUBTYPE AND CL=VARID THEN                     00856100 P  105
$! BY KFK                                                                          P  105
$! DATE 11/31/74                                                                    P  105
$! THIS PATCH CORRECTS AN ERROR THE OCCURS IF A COMPLEX OR                         P  105
$! DOUBLE PRECISION FUNCTION IS PASSED AS A PARAMETER                             P  105
$! TO A FUNCTION OR SUBROUTINE.  THE PROGRAM COULD GET                            P  105
$! DS-ED FOR INVALID INDEX OR STACK OVERFLOW AND SOMETIMES SYSTEM HANGS            P  105
$! COULD OCCUR.                                                                   P  105


$#PATCH NUMBER 106 FOR FORTRAN CONTAINS 2 CARDS.                                       P  106

          END;                                                          00813000 P  106
    IF NPARMS > 0 THEN                                                  00816000 P  106
$! THIS PATCH ELIMINATES SOME GARBAGE ON THE SYMBOL FILE                           P  106
$! BY DDD                                                                          P  106
$! 3/4/75                                                                         P  106
```

```
$!*************************************************************************          P  106



$#PATCH NUMBER 107 FOR FORTRAN CONTAINS 1 CARD.                                      P  107

              BUF ← SCANINC(BUF,ID,RESULT,0,1);                          02364420 P  107
$! THIS PATCH WILL CORRECT A PROBLEM WITH "$ INCLUDE" NOT WORKING.                  P  107
$! BY DDD                                                                          P  107
$! DATE 12/27/74                                                                   P  107
$!*************************************************************************       P  107



$#PATCH NUMBER 108 FOR FORTRAN CONTAINS 1 CARD                                       P  108

     IF A > REALTYPE OR B > REALTYPE THEN                                  03977000 P  108
$!BY KFK                                                                           P  108
$!DATE 2/8/75                                                                      P  108
$!SOFTWARE FLASH #                                                                 P  108
$!  THIS CHANGE WILL ALLOW THE FORTRAN COMPILER TO CORRECTLY SYNTAX                 P  108
$!CALLS ON FUNCTION STATEMENTS. WITHOUT THIS PATCH THE COMPILER                     P  108
$!WOULD NOT GIVE A SYNTAX ERROR IF THE ACTUAL PARAMETER IS A DOUBLE                P  108
$!PRECISION VARIABLE AND THE FORMAL PARAMETER IS A REAL VARIABLE.                   P  108
$!THUS THE STACK WOULD BE SET UP WRONG AND THE OBJECT PROGRAM COULD                 P  108
$!BE DS-ED FOR INVALID ADDRESS OR STACK OVERFLOW, AND SOMETIMES                     P  108
$!SYSTEM HALTS COULD HAPPEN. ALSO IF A REAL VARIABLE WERE PASSED                   P  108
$!TO A INTEGER FORMAL PARAMETER AN INVALID SYNTAX ERROR WOULD BE GIVEN.            P  108
$!    NOTE:  THE FORMAL PARAMETERS, SOMETIMES CALLED DUMMY VARIABLES               P  108
$!CAN BE GIVEN A TYPE, OTHER THAN THE DEFAULT TYPE, BY                              P  108
$!REFERENCE TO THEM IN A TYPE DECLARATIVE STATEMENT OR AN                           P  108
$!IMPLICIT TYPE STATEMENT.                                                         P  108
$!EXAMPLE:                                                                         P  108
$!     REAL *8 X,F, B      IN THIS EXAMPLE THE DOUBLE PRECISION                    P  108
$!     F(X) = X*X       FUNCTION F HAS ONE DOUBLE PRECISION                        P  108
$!     A = SNGL(F(B))    PARAMETER B.                                             P  108
$!     STOP                                                                       P  108
$!     END                                                                        P  108
$!*************************************************************************       P  108



$#PATCH NUMBER 109 FOR FORTRAN CONTAINS 7 CARDS.                                     P  109

    BOOLEAN VARF, SINGLETOG;                                           03791000 P  109
   IF LINK < 0 THEN BEGIN SINGLETOG ← TRUE; LINK ← ABS(LINK) END;      03799500 P  109
    IF NOT SINGLETOG AND INFA.SUBCLASS > LOGTYPE THEN                  03834000 P  109
        BEGIN LENGTH ← 2 × LENGTH; BOUNDS ← TRUE END;                  03834500 P  109
    BOOLEAN DOUBLED, SINGLETOG;                                        05066005 P  109
                     SINGLETOG ← TRUE;                                 05067275 P  109
        FX1 ← IF SINGLETOG THEN -FNEXT ELSE FNEXT;                     05069000 P  109
$!BY DDD MSA-CENTRAL                                                               P  109
$!DATE 2/14/75                                                                     P  109
$!THIS PATCH CORRECTS A PROBLEM IN THE IMPLEMENTATION                              P  109
```

```
$! OF THE IMPLICIT CONSTRUCT.                                                  P  109
$!*************************************************************            P  109



$#PATCH NUMBER 110 FOR FORTRAN CONTAINS 2 CARDS.                               P  110

  IF NOT(LASTNEXT=42 OR LASTNEXT=1000 OR LASTNEXT=30                  05529130 P  110
         OR LASTNEXT=16 OR LASTNEXT = 11)                             05529131 P  110
$! DDD                                                                        P  110
$! 3/2/75                                                                    P  110
$! THIS PATCH ALLOWS SUBROUTINES TO BE PLACED BEFORE IMPLICIT STATEMENTS     P  110
$! THAT ARE DECLARED IN THE MAIN PROGRAM.                                    P  110
$!*************************************************************            P  110



$#PATCH NUMBER 111 FOR FORTRAN CONTAINS 5 CARDS. ILLEGAL * IN RUN-TIME EDITING  C  111

$! CORRECTS THE ERROR MESSAGE PRINTED WHEN AN ASTERISK IS USED IMPROPERLY AS    C  111
$! PART OF A FORMAT PHRASE.                                                    C  111
$! BY ARO - MSA CENTRAL SOFTWARE FLASH 169                                     C  111
183   ILLEGAL USE OF ASTERISK FOR RUN-TIME EDITING                    00143043 C  111
 "ILLEGAL ","USE OF A","STERISK ","FOR RUN-","TIME EDI","TING    ",   00596535 C  111
             BEGIN FLAG(183) ;                                       04708515 C  111
          IF ZF OR DECIMAL NEQ 0 THEN FLAG(183); DECIMAL:=4095;      04735715 C  111
          END ELSE BEGIN DECIMAL:=4095; FLAG(183); GO FL END ;       04735750 C  111
$!*************************************************************            C  111
```

```
$#PATCH NUMBER 112 FOR FORTRAN CONTAINS 14 CARDS. EXTRA OR MISSING "END" STMTS.  C  112

$! THE FOLLOWING PATCH CORRECTS THE HANDLING OF MULTIPLE "END" STATEMENTS.      C  112
$! PREVIOUSLY, ALL THE REGULAR END-OF-SEGMENT PROCESSING WAS DONE ON            C  112
$! SUPERFLUOUS "ENDS.  THIS COULD CAUSE A "DEC ERROR" AND AN ABORTED COMPILE IF C  112
$! A CROSS REFERENCE WAS REQUESTED.  NOW A MESSAGE IS PRINTED AND NO END-OF-    C  112
$! SEGMENT PROCESSING IS DONE FOR SUPERFLUOUS "END"S.  THIS PATCH ALSO CORRECTS C  112
$! THE HANDLING OF MISSING "END" STATEMENTS.                                    C  112
$! BY ARO - MSA CENTRAL SOFTWARE FLASH 170                                     C  112
184   NO PROGRAM UNIT FOR THIS END STATEMENT                        00143044 C  112
 "NO PROGR","AM UNIT ","FOR THIS"," END STA","TEMENT  ","        ",  00596536 C  112
          IF SPLINK NEQ 0 THEN BEGIN FLAG(5); ENDS; SEGMENTSTART; END; 05128000 C  112
          IF SPLINK NEQ 0 THEN BEGIN FLAG(5); ENDS; SEGMENTSTART; END; 05151000 C  112
        IF SPLINK=0 THEN FLAG(184) ELSE                               05212005 C  112
        BEGIN                                                         05212007 C  112
        END;                                                          05216000 C  112
          IF SPLINK NEQ 0 THEN BEGIN FLAG(5); ENDS; SEGMENTSTART; END; 05532000 C  112
       BOOLEAN ENDTOG;                                                05537200 C  112
          BEGIN ENDS; ENDTOG:=TRUE; SCAN END;                         05553000 C  112
          IF NOT ENDTOG THEN IF SPLINK=0 THEN SPLINK:=1;              05579000 C  112
          ENDTOG:=FALSE;                                              05579100 C  112
  IF NOT ENDSEGTOG THEN IF SPLINK NEQ 0                               05612100 C  112
     THEN BEGIN XTA:=BLANKS; FLAG(5); ENDS END;                       05612200 C  112
```

```
$#PATCH NUMBER 113 FOR FORTRAN CONTAINS 7 CARDS. FIX X**I                          C  113

$! THE FOLLOWING PATCH CORRECTS THE CODE GENERATED FOR X TO THE I POWER, WHERE    C  113
$! I IS AN INTEGER CONSTANT GREATER THAN 1023 WHOSE BINARY REPRESENTATION ENDS    C  113
$! IN 2 ZEROES.  PREVIOUSLY, AN EXPONENT OVERFLOW OCCURRED.                       C  113
$! BY ARO AND JTC   MSA CENTRAL   SOFTWARE FLASH 171                            C  113
$! HERIOT-WATT U. PATCH 515 CLAIMS TO FIX BUGS IN THIS PATCH.  INVESTIGATE AND   C  113
$! INSTALL HWU 515.  * * * * * * * * * * * * * * * * * * * *                    C  113
   BOOLEAN CNSTSEENLAST;       %FOR HANDLING CONSTANT                 04025500 C  113
   REAL SAVEADR;               %EXPONENTS                             04025600 C  113
   CNSTSEENLAST:=FALSE;                                               04072500 C  113
   SAVEADR:=ADR; CNSTSEENLAST:=TRUE;                                  04178500 C  113
      CNSTSEENLAST THEN                                               04345000 C  113
      EXPV:=LINK;                                                     04349000 C  113
      A:=1; ADR:=SAVEADR;                                             04350000 C  113
$!*************************************************************************    C  113


$#PATCH NUMBER 114 FOR FORTRAN CONTAINS 5 CARDS. FIX $VOIDT                       C  114

$! THE FOLLOWING PATCH CORRECTS A $VOIDT PROBLEM.  PREVIOUSLY, THE LAST CARD     C  114
$! IMAGE IN THE "TAPE" FILE COULD NOT BE VOIDED.                                 C  114
$! BY ARO - MSA CENTRAL SOFTWARE FLASH 172                                       C  114
$! SEE ALSO PATCH 115                                                            C  114
    IF VOIDTTOG THEN IF SEQCHK(CRD[9],VOIDTSEQ) < 4                    02373045 C  114
         THEN VOIDTTOG:=FALSE                                          02373100 C  114
          ELSE BEGIN VOIDTTOG:=FALSE; GO TO STRT; END;                 02373200 C  114
          VOIDTTOG:=FALSE ELSE BEGIN NEXTCARD:=SEQCHK(TB[9],CB[9]);    02416200 C  114
                                GO TO STRT; END;                       02416300 C  114
$!*************************************************************************    C  114


$#PATCH NUMBER 115 FOR FORTRAN CONTAINS 3 CARDS. FIX $VOIDT                       C  115

$! THE FOLLOWING PATCH CORRECTS A PROBLEM WHICH CAUSED THE WRONG RECORDS TO BE   C  115
$! VOIDED WHEN THE VOIDING SEQUENCE LIMIT ON THE $VOID OR $VOIDT CARD IMAGE      C  115
$! WAS LESS THAN 8 DIGITS LONG.  NOW THE INTENDED RECORDS ARE VOIDED IN ALL      C  115
$! CASES.                                                                       C  115
$! BY ARO - MSA CENTRAL SOFTWARE FLASH 173                                       C  115
$! SEE ALSO PATCH 114                                                            C  115
         DS:=8LIT"0";                                                 02252600 C  115
         8(SI:=SI+1; DI:=DI+1; TALLY:=TALLY+1;                        02252800 C  115
           IF SC LSS "0" THEN JUMP OUT);                              02252850 C  115
$!*************************************************************************    C  115


$# PATCH 400 FOR FORTRAN.XVI.0 CONTAINS 9 CARDS. FIX $INCLUDE                    C  400
```

```
$! THIS PATCH CORRECTS THE $ INCLUDE HANDLING.  THE $ INCLUDE SCANNER              C  400
$! RETURNED A RESULT OF REQUESTED TYPE INSTEAD OF FOUND TYPE (I.E. YOU             C  400
$! WERE SCANNING FOR A NUMBER THUS YOU GOT A NUMBER).  IF A STARTING SEQ           C  400
$! WAS GIVEN THEN THE COMPILER WOULD INSIST ON A SPECIAL CHARACTER                 C  400
$! FOLLOWED BY A NUMBER.  THIS WOULD CAUSE PROBLEMS IF ONLY THE STARTING          C  400
$! SEQUENCE WAS SPECIFIED,  THE SPECIAL CHARACTER FOUND WOULD BE THE SCAN STOP    C  400
$! IN CC 73 AND THE NUMBER WOULD BE THE LAST 7 DIGITS OF THE SEQUENCE             C  400
$! NUMBER.  IF THE SEQUENCE NUMBER WERE BLANK, THE COMPILER WOULD CONVERT         C  400
$! "]     " TO DECIMAL AS THE ENDING SEQUENCE NUMBER.  IN EITHER CASE,           C  400
$! THE RESULTS OF SPECIFYING ONLY THE STARTING SEQUENCE NUMBER WERE              C  400
$! UNPREDICTABLE.                                                               C  400
$! BY D.G. WILLMS, 17FEB1976     BROCK U. PATCH 400                              C  400
        IF SC = ALPHA THEN ELSE BEGIN DS:=LIT "1"; DI:=ID; DS:=2LIT"0";02333240 C  400
        IF SC LSS "0" THEN DS:=LIT "2" ELSE DS:=LIT "3";                 02333270 C  400
        N (DI:=ID; DS:=8 LIT "0       "; DI:=DI-7;                       02333280 C  400
        M ( 8 ( IF SC LSS "0" THEN JUMP OUT 2 TO LQ;                     02333360 C  400
                IF SC GTR "9" THEN JUMP OUT 2 TO LQ;                     02333370 C  400
             SI:=SI+1; TALLY:=TALLY+1));                                 02333380 C  400
               IF ID = "]     " THEN NEWSEQ (INSERTSEQ,99999999)         02364605 C  400
               ELSE BEGIN                                                02364610 C  400
               END                                                       02364670 C  400
$!*************************************************************************     C  400


$# PATCH 501 FOR FORTRAN.XV.1 CONTAINS 14 CARDS. FIX UP & CHANGE $-CARDS        C  501

$! DEFAULT $-CARD IS:  $CARD LIST SINGLE -CHECK -PAGE                           C  501
$! SINGLE AND CHECK ALWAYS SET ON A $-CARD AND CAN ONLY BE RESET                C  501
$! BY USING THE EXTENDED $-CARD "RESET"                                         C  501
$! ALLOW NEW TAPE OPTION TO BE "NEW", "NEWTAPE", OR "NEW TAPE"                  C  501
$! BY DGW&GRK, BROCK U, PATCH 501                                               C  501
$! MODIFIED BY DR, UCSC, APRIL 7, 1977 SO THAT THE DEFAULT ACTION IS NOT        C  501
$! TO CHECK SEQUENCE NUMBERS                                                    C  501
$! SEE ALSO PATCHES 511, 517                                                    C  501
$! SEE ALSO HERIOT-WATT U, PATCH 702                                           C  501
        TSSMESTOG ← VAL; TSSEDITOG ← VAL; CHECKTOG ← VAL;            02269300 C  501
        SINGLETOG←NOT VAL;  HOLTOG←VAL;                              02269600 C  501
        PRTOG←VAL;  DOLIST←VAL;  LIBTAPE←VAL;  SEGPTOG←VAL;          02270200 C  501
           BEGIN LIBTAPE←VAL; NEWTPTOG←VAL; NTAPTOG←TRUE;            02275100 C  501
           IF ID = "NEW   " THEN                                     02275120 C  501
              BEGIN                                                  02275140 C  501
              GETID;                                                 02275160 C  501
              IF ID ≠ "TAPE  " THEN                                  02275180 C  501
                   GO TO LP;                                         02275200 C  501
              END;                                                   02275220 C  501
           END ELSE                                                  02275240 C  501
        IF ID = "NEWPAG" THEN SEGPTOG ← VAL ELSE                     02286100 C  501
LISTOG←TRUE;  SINGLETOG←TRUE; CHECKTOG ← FALSE;  %DEFAULT            03384000 C  501
SEGPTOG ← FALSE;   %INHIBIT PAGE SKIP AFTER SUBROUTINES              03571300 C  501


$# PATCH 502 FOR FORTRAN.XV.3 CONTAINS 2 CARDS. CHANGE TAPE LOCK CONDITIONS    C  502
```

```
$! LOCK TAPE FILE BEFORE NEWTAPE FILE INSTEAD OF CLOSING ONLY IF                    C  502
$! A NEW FILE IS CREATED.                                                          C  502
$! D G WILLMS, 19DEC74     BROCK U, PATCH 502                                      C  502
        LOCK(TAPE);       %RW/L TAPE FILE OR LOCK DISK                 03366100 C  502
        IF NTAPTOG THEN LOCK(NEWTAPE,*); %RW/L TAPE OR CRUNCH DISK     03366200 C  502



$# PATCH 503 FOR FORTRAN.XII CONTAINS 13 CARDS. DEFAULT FILE NAMES ETC            C  503

$! SET FILE 5 TO READER AND FILE 6 TO BE PRINTER BY DEFAULT                       C  503
$! BY KEN WEST       BROCK U. PATCH 503                                          C  503
 LABEL QQQDISKDEFAULT;                                          02997010 C  503
 QQQDISKDEFAULT:                                                03164045 C  503
 IF IDNM="READER " OR IDNM="FILE5  " THEN IDNM←"CARD   " ELSE    03184010 C  503
 IF IDNM="FILE6  " THEN BEGIN IDNM←"PRINTER";FILTYP←18;END ELSE  03184020 C  503
  BEGIN                                                         03184030 C  503
  EMITL(20);  EMITL(600);  FILTYP←12;  %20 x 600 REC DISK       03184040 C  503
     J←0&2[42:42:6]&10[30:36:12]&300[18:36:12];                 03184050 C  503
  FID←IDNM;  MFID←"FORTEMP";  T1←GITSZ(IDNM);                   03184054 C  503
      GO TO QQQDISKDEFAULT;                                     03184060 C  503
  END;                                                          03184070 C  503
   ELSE EMITDESCLIT(FILECHECK(",5    ",2+17×REAL                04841000 C  503
     ELSE IF N = 2 THEN                                         04844000 C  503
        EMITDESCLIT(FILECHECK(",6    ",2+17×REAL               04845000 C  503
$!*************************************************************          C  503
```

```
$# PATCH 510 FOR FORTRAN.XIII CONTAINS 4 CARDS. NO ERR MSG IF NO ERRS             C  510

$! IF NO SEQUENCE ERRORS THEN LINE IS BLANKED OUT                                C  510
$! FIXES "NUMBER OF SEQUENCE ERRORS" LINE IN COMPILER WRAP UP.                    C  510
$! BROCK U. PATCH 510                                                            C  510
       DS←32 LIT "                                  ";                 05603081 C  510
       DS←32 LIT "                                  ";                 05603082 C  510
   WRITE(RITE,FOC1,FIELD(ERRORCT),ERRORCT,IF SEQERRCT=0 THEN 99 ELSE    05617000 C  510
   IF ERRORCT>0 THEN IF FLAGLAST(ERRORBUFF,LASTERR) THEN WRITE(RITE,15, 05619050 C  510
$!*************************************************************          C  510



$# PATCH 511 FOR FORTRAN.XVI.0 CONTAINS 6 CARDS. LIST $ CARDS                     C  511

$! THIS PATCH CAUSES THE COMPILER TO LIST $ CARDS.  THE $ CARDS WILL BE LISTED   C  511
$! IF LIST OR LISTDOLLAR IS SET.  THE $ CARDS WILL ALSO BE LISTED IF THE         C  511
$! $ CARD IS SETTING LIST FALSE AND THE LIST OPTION WAS PREVIOUSLY SET.          C  511
$! $ INCLUDE CARDS WILL BE LISTED IF LIST OR LISTDOLLAR IS SPECIFIED.           C  511
$! THE $ INCLUDE CARD WILL BE LISTED (IF LIST OR LISTDOLLAR IS SET) IF COPY     C  511
$! IS SPECIFIED.  PREVIOUSLY THE $ INCLUDE CARD WAS NOT LISTED IF COPY          C  511
$! WAS SPECIFIED.                                                               C  511
$! D.G. WILLMS, 19DEC74. REWRITTEN BY D.G. WILLMS, 20AUG75.                     C  511
$! BROCK U. PATCH 511                                                           C  511
```

```
$: MODIFIED BY DR, UCSC, MAY 10, 1977 TO ELIMINATE ALL REFERENCES TO "IGNORE"     C  511
$: SEE ALSO PATCHES 501, 517                                                      C  511
    BOOLEAN VAL, SAVELISTOG;                                          02261500 C  511
        SAVELISTOG := LISTOG;                                         02262775 C  511
        IF DOLIST OR LISTOG OR SAVELISTOG THEN                        02289450 C  511
           PRINTCARD;    %WAS CONDITIONED BY "DOLIST"          GRK    02289500 C  511
        END;                                                          02364885 C  511
        IF LISTOG OR DOLIST THEN PRINTCARD;                           02364890 C  511
$:*************************************************************************** C  511



$# PATCH 512 FOR FORTRAN CONTAINS 1 CARD, BETTER CORE ESTIMATE                    C  512

$: PRODUCE MORE ACCURATE CORE ESTIMATES IF ARRAYS ARE USED                       C  512
$: BY GRK      BROCK U, PATCH 512                                                C  512
        LNK := 256;   %INCLUDE ENTIRE ARRAY SIZE IN ESTIMATE          01719000 C  512
$:*************************************************************************** C  512



$# PATCH 515 FOR FORTRAN CONTAINS 1 CARD, SINGLE SPACE COMPILER SUMMARY           C  515

$: BY G.R. KENNEDY   08FEB73      BROCK U, PATCH 515                             C  515
DEFINE PTR=LINE#,RITE=LINE#,CR=CARD#,TP=TAPE#;                    00391000 C  515
$:*************************************************************************** C  515
```

```
$# PATCH 517 FOR FORTRAN.XV.1 CONTAINS 5 CARDS, $-CARD FIX                       C  517

$: THIS PATCH WILL ALLOW THE COMPILER TO CORRECTLY HANDLE $ CARDS                C  517
$: SUCH AS $CARD RESET CHECK.  PRIOR TO THIS CHANGE, THE COMPILER                C  517
$: WOULD NOT SET ANY OPTIONS TO THEIR DEFAULT STATUS IF THE FIRST                C  517
$: OPTION WAS "CARD" OR "TAPE" AND THE SECOND OPTION WAS "SET" OR "RESET"        C  517
$: THIS PATCH ALSO PREVENTS A DOLLAR CARD WITH AN ILLEGAL CONSTRUCT FROM         C  517
$: BEING PRINTED MORE THAN ONCE.                                                 C  517
$: BY DGW     BROCK U, PATCH 517                                                 C  517
$: MODIFIED BY DR, UCSC APRIL 6, 1977, TO REMOVE A BROCK U. FEATURE THAT         C  517
$: PERMITTED TRACE, PRT, AND DEBUGN ONLY TO SYSTEMS PROGRAMMERS                  C  517
$: SEE ALSO PATCHES 501, 511                                                     C  517
          TAPETOG.[47:1] := LASTMODE = 2;                         02265700 C  517
$VOIDT 02267200                                                   02266000 C  517
            BEGIN TAPETOG.[47:1]:=NOT VAL; LASTMODE:=2-REAL(VAL) END ELSE 02272510 C  517
               IF UNPRINTED THEN BEGIN PRINTCARD; UNPRINTED:=FALSE END;   02287000 C  517
            IF UNPRINTED THEN PRINTCARD;                          02289500 C  517



$# PATCH 992 FOR FORTRAN CONTAINS 1 CARD,  COMMENTS                              C  992

$: THIS PATCH MERELY ADDS COMMENTS TO EXISTING CODE.                             C  992
$: BY DR  UCSC    APRIL 18, 1977                                                 C  992
```

```
      IF INFA.CLASS = FILFID THEN   %SEE COMMENTS ON LINE 02118000        03144000 C  992



$# PATCH 993 FOR FORTRAN CONTAINS 21 CARDS.  COMMENTS                              C  993

$: THIS PATCH MERELY ADDS COMMENTS TO EXISTING CODE.                               C  993
$: BY DR  UCSC    APRIL 18, 1977                                                   C  993
GO CASEL[-IDINFO];    % SEE INITIALIZATION OF "TIPE", LINE 03433100      02862800 C  993
L1:       %DIGITS                                                       02863000 C  993
L2:       % >                                                           02898100 C  993
L3:       % ≥                                                           02899100 C  993
L4:       % & OR +                                                      02900100 C  993
L5:       % .                                                           02910100 C  993
L6:       % % OR (                                                      02929100 C  993
L7:       % <                                                           02930100 C  993
L8:       % LETTER O                                                    02938100 C  993
L9:       % $                                                           02942100 C  993
L10:      % *                                                           02943100 C  993
$: L11 IS ALTERED IN PATCH 990                                                     C  993
L12:      % -                                                           02946100 C  993
L13:      % ) OR [                                                      02947100 C  993
L14:      % ;                                                           02948100 C  993
L15:      % <                                                           02949100 C  993
L16:      % /                                                           02951100 C  993
L17:      % ,                                                           02960100 C  993
L18:      % ≠                                                           02962100 C  993
L19:      % = OR ← OR #                                                 02963100 C  993
L20:      % ]                                                           02964100 C  993
L21:      % " OR : OR @                                                 02965100 C  993
$:*************************************************************************** C  993



$# PATCH 994 FOR FORTRAN CONTAINS 14 CARDS. LABELS FOR CASES                       C  994

$: THIS PATCH INTRODUCES LABELS ON THE VARIOUS CASES OF A CASE STATEMENT,          C  994
$: THESE LABELS MERELY HELP IDENTIFY THE CASES TO A HUMAN READER OF THE LISTING.  C  994
$: THOSE CASES WHICH NEED LABELS FOR PROGRAMMING PURPOSES ALREADY HAVE BEEN       C  994
$: LABELLED.                                                                       C  994
$: BY DR  UCSC    APRIL 18, 1977                                                   C  994
LABEL LOOP, CASESTMT;                                                 02803000 C  994
LABEL CASE0,CASE1,CASE2,CASE3,CASE4,CASE5,CASE6,CASE7;                02803100 C  994
LABEL CASE8,CASE9,CASE10,CASE11,CASE12,CASE13,CASE14;                 02803200 C  994
CASE0:                                                                02807999 C  994
CASE4:                                                                02827999 C  994
CASE6:                                                                02830999 C  994
CASE7:                                                                02833999 C  994
CASE8:                                                                02836999 C  994
CASE9:                                                                02837999 C  994
CASE10:                                                               02838999 C  994
CASE11:                                                               02839999 C  994
CASE12:                                                               02840999 C  994
CASE13:                                                               02848999 C  994
CASE14:                                                               02849999 C  994
```

```
$# PATCH 995 FOR FORTRAN CONTAINS 1 CARD. ELIMINATE DUPLICATE FORWARD REFERENCES C  995

$! BY DR  UCSC    APRIL 18, 1977                                                C  995
$! LINE 02057000 = LINE 00420000                                               C  995
$ VOIDT                                                              02057000 C  995



$# PATCH 996 FOR FORTRAN CONTAINS 41 CARDS.  COMMENTS                          C  996

$! THIS PATCH MERELY ADDS COMMENTS TO EXISTING CODE                            C  996
$! BY DR  UCSC    APRIL 18, 1977                                               C  996
DEFINE CE       = [2:1]#,          %INFA                             00243000 C  996
       LASTC    = [3:12]#,         %INFA                             00244000 C  996
       SEGNO    = [3:9]#,          %INFA                             00245000 C  996
       CLASS    = [15:5]#,         %INFA                             00246000 C  996
       EQ       = [20:1]#,         %INFA                             00246100 C  996
       SUBCLASS = [21:3]#,         %INFA                             00247000 C  996
       CLASNSUB = [14:10]#,        %INFA                             00248000 C  996
       ADJ      = [14:1]#,         %INFA                             00249000 C  996
       TWOD     = [14:1]#,         %INFA                             00250000 C  996
       FORMAL   = [13:1]#,         %INFA                             00251000 C  996
       TYPEFIXED= [12:1]#,         %INFA                             00252000 C  996
       ADDR     = [24:12]#,        %INFA                             00253000 C  996
       LINK     = [36:12]#,        %INFC                             00254000 C  996
       BASE     = [18:15]#,        %INFC                             00255000 C  996
       SIZE     = [33:15]#,        %INFC                             00256000 C  996
       BASENSIZE= [18:30]#,        %INFC                             00257000 C  996
       NEXTRA   = [2:6]#,          %INFC                             00258000 C  996
       ADINFO   = [8:10]#,         %INFC                             00259000 C  996
       RELADD   = [21:15]#,        %INFA                             00260000 C  996
       TOCE     = 2:47:1#,         %INFA                             00261000 C  996
       TOSEGNO  = 3:39:9#,         %INFA                             00262000 C  996
       TOLASTC  = 3:36:12#,        %INFA                             00262100 C  996
       TOCLASS  = 15:43:5#,        %INFA                             00263000 C  996
       TOEQ     = 20:47:1#,        %INFA                             00263100 C  996
       TOSUBCL  = 21:45:3#,        %INFA                             00264000 C  996
       TOADJ    = 14:47:1#,        %INFA                             00265000 C  996
       TOFORMAL = 13:47:1#,        %INFA                             00266000 C  996
       TOTYPE   = 12:47:1#,        %INFA                             00267000 C  996
       TOADDR   = 24:36:12#,       %INFA                             00268000 C  996
       TOLINK   = 36:36:12#,       %INFC                             00269000 C  996
       TOBASE   = 18:33:15#,       %INFC                             00270000 C  996
       TOSIZE   = 33:33:15#,       %INFC                             00271000 C  996
       TONEXTRA = 2:42:6#,         %INFC                             00272000 C  996
       TOADINFO = 8:38:10#,        %INFC                             00273000 C  996
       TORELADD = 21:33:15#,       %INFA                             00274000 C  996
%THE FOLLOWING CLASSES APPEAR IN THE 1ST WORD OF EACH 3-WORD ENTRY    00274998 C  996
%OF THE "INFO" ARRAY, AND MAY BE COPIED TO INFA.CLASS.               00274999 C  996
   JUNK     = 17#,                 %NOT A GENUINE CLASS              00290055 C  996
%THE FOLLOWING SUBCLASSES APPEAR IN THE 1ST WORD OF SOME 3-WORD      00294998 C  996
%ENTRIES OF THE "INFO" ARRAY, AND MAY BE COPIED TO INFA.SUBCLASS.    00294999 C  996
```

```
DEFINE THRU = STEP 1 UNTIL #;                                          00316000 C  996



$# PATCH 997 FOR FORTRAN CONTAINS 5 CARDS.  WARNING                              C  997

% BEWARE ALL YE WHO SEEK HEREIN:                                       00001100 C  997
% THIS COMPILER IS A MESS.  THE CONCEPTS OF CHARACTER CONVERSION FROM   00001105 C  997
% HOLLERITH (= EBCDIC), SCANNING, AND PARSING ARE NOT CLEARLY SEPARATED 00001110 C  997
% IN THE CODE.  FOR EXAMPLE, PROCEDURE "SCAN" TRIES TO DO ALL 3 AT ONCE.00001115 C  997
% DAN ROSS    UCSC    APRIL 12, 1977                                    00001120 C  997



$# PATCH 998 FOR FORTRAN CONTAINS 1 CARD, UNLABELLED PAPER TAPE                  C  998

$: THIS PATCH MAKES "UNIT=PAPER" VALID ON A FILE CARD.  IT WILL GENERATE A FILE  C  998
$: PARAMETER BLOCK WITH UNIT=8, WHICH MEANS UNLABELLED PAPER TAPE.  INPUT AND    C  998
$: OUTPUT ARE DISTINGUISHED BY THE FILE NAME, WHICH MUST BE EITHER               C  998
$: "PRA/PRA" FOR INPUT OR "PPA/PPA" FOR OUTPUT.  CAUTION:  USERS MUST NOT MAKE   C  998
$: MISTAKES ON FILE CARDS.  THE CODE FOR PARSING FILE CARDS IS UNSTABLE, AND IT  C  998
$: WOULD BE A MAJOR RECODING EFFORT TO MAKE IT STABLE.                           C  998
$: BY DR, UCSC   APRIL 7, 1977                                                   C  998
$: SEE ALSO HERIOT-WATT U. PATCH 720                                             C  998
                          IF TOG + XTA = "PAPER " THEN 8 ELSE          02214100 C  998



$#PATCH NUMBER 999 FOR FORTRAN CONTAINS 1 CARDS.  PATCH LEVEL                   C  999

     "16"                                                              02025000 C  999
$:*************************************************************************** C  999
```

```
**********    CONFLICTS    ***************************************************************************************

          IF UNPRINTED THEN PRINTCARD;                                    02289500 C  517 CONFLICTED WITH:
          PRINTCARD;    %WAS CONDITIONED BY "DOLIST"          GRK       02289500 C  511    DISCARDED

        LOCK(TAPE);       %RW/L TAPE FILE OR LOCK DISK                  03366100 C  502 CONFLICTED WITH:
IF NTAPTOG THEN BEGIN CLOSE(TAPE);%CLOSE TAPE BEFORE NEWTAPE            03366100 C  102    DISCARDED

       IF NTAPTOG THEN LOCK(NEWTAPE,*); %RW/L TAPE OR CRUNCH DISK       03366200 C  502 CONFLICTED WITH:
    LOCK(NEWTAPE,*); END;                                               03366200 C  102    DISCARDED



NUMBER OF ERRORS DETECTED =  0.
PROCESSOR TIME =   16 SECONDS.
I/O TIME =   18 SECONDS.

 LABEL  000000000LINE   00177202?EXECUTE PATCH/MERGE                                        PATCH  /MERGE
```