```
0002                    version ==      2
0003                    revision==      3
                                .pabs
                                .phex
                    ;
                    ;-----------------------------------
                    ;
                    ;       Multiple floppy disk test program
                    ;
                    ;       This program tests up to eight drives. There are
                    ; 3 write patterns:  Linear,Random, and fixed.  After
                    ; a write has been performed on all drives, each drive
                    ; read tested in two successive patterns:  Linear and
                    ; Random.  The user selects the density, as well as the
                    ; order in which the drives are tested.  Control-C will
                    ; abort the program at any time.  Console 'in progress'
                    ; commands are F,R,S,T, and Q and are described below.
                    ;
                    ; F -  Initiate FIXTEST.  This turns off the disk-write
                    ;       (if its enabled), sets the test pattern to fixed,
                    ;       and enables the user to Read test or Write test
                    ;       at will to any disk, any track, for any length of
                    ;       time.
                    ;
                    ; R  - Read out the 'error' buffer.  The program keeps
                    ;       track of all write patterns and loop numbers, by
                    ;       writing them to a buffer located at 7000h. All
                    ;       errors are written to this buffer, also. When
                    ;       the buffer fills past B000h the buffer is des-
                    ;       troyed and re-initialized to 7000h.
                    ;
                    ; S  - Print the current status to console. This in-
                    ;       the user immediately as to what function is
                    ;       currently being performed, the amount of errors
                    ;       that have been detected, as well as the current
                    ;       drive and current time
                    ;
                    ; T  - Stop the current test in progress and select
                    ;       a new set of disks to test.  The buffer-write
                    ;       disk is kept the same, and the buffer-file's
                    ;       contents are not destroyed.
                    ;
                    ; Q  - Quit the program.  The program then asks the
                    ;       user if an exit is really desired.  If no 'Y'
                    ;       is entered, the test resumes as if not inter-
                    ;       rupted.  If a 'Y' is entered the test will
                    ;       then terminate.
                    ;
                    ; ?  - Any other character is received from the console
                    ;       a menu of acceptable commands is printed out.
                    ;
                    ; Flexibilities
                    ;
                    ;       Changeable fixed byte. Change 'wrstbyt'.
                    ; If a change of the worst byte ( read and
```

```
                        ; written in the 'fixed' mode ) is desired,
                        ; change the value of the byte at 103h to the
                        ; desired value.  This can be done in ZDT if desired.
                        ;
                        ;       Changeable error overflow.  Change 'maxerrs'.
                        ; If any drive accumulates a 'maxerrs' number of errors
                        ; of any one type, FDTEST will automatically log that
                        ; drive out of the test.  'Maxerrs' is defaulted to 10
                        ; and can be set by changing the byte at 104h to any
                        ; desired value.  This also can be done through ZDT.
                        ;
                        ;       Disk write option.  This writes the contents
                        ; of the error buffer ( which begins at 7000h ) to
                        ; a user-selected disk. When 128 bytes are collected
                        ; the sectors worth of data is written to CONFILE.TXT
                        ; (this file is chosen by default if no file name is
                        ; found on the command line) and the buffer is cleared.
                        ; This will continue forever, or until the disk is
                        ; full.
                        ;
                        ;       Fixed track testing.  See 'F' command above.
                        ;
                        ; Please note: Both CP/M and BIOS calls are used.
                        ; The disk write feature uses CP/M file management,
                        ; while the test reads & writes are performed through
                        ; BIOS calls.
                        ;
                        ;                                       David Stein 12/10/80
                        ;
                        ;  Update History:
                        ;
                        ;       Version 2.03 - Condensed error buffer info.
                        ;                      Neatened error messages.
                        ;                                              DS  3/26/81
0100                            .loc    100h
0100    C3 0105                 jmp     start

                        ; Patch area
0103    B6              wrstbyt:.byte   0B6h    ; special worst case test
                                                ;byte for fixed pattern reading
0104    0A              maxerrs:.byte   0Ah     ; Maximum number of Sector or
                                                ;Track errors allowed on each
                                                ;drive before logging drive out

                        ; Define some constants:
000D                    cr      =       0Dh     ; define carraige return
000A                    lf      =       0Ah     ; define line feed
000A                    retry   =       10      ; number of error retries
004E                    BIOSrty =       4Eh     ; memory location of # retries
0080                    comline =       80h     ; CP/M command line location
005C                    DFCB    =       5Ch     ; CP/M com line FCB addr
                        ;
                        ;----------
                        ; Main Program
0105                    start:
```

```
                        ; Start with a fresh stack, initialize buffer and flags
0105    31 7000                 lxi     sp,stack
0108    3E00                    mvi     A,0
010A    32 1537                 sta     errflg      ; turn off error flag
010D    32 162B                 sta     dskflg      ; turn off disk write flag
0110    21 7000                 lxi     H,erbuffer
0113    22 1535                 shld    errbufadr ; initialize err buffer
0116    CD 170F                 call    putEOF      ; label end of err buffer
                        ; Check if command line filename option was used.
0119    21 1720                 lxi     H,FCB
011C    22 171E                 shld    FCBaddr     ; Load in FCB for CONFILE.TXT
011F    3A 0080                 lda     comline     ; Now we check the com line
0122    FE03                    cpi     3           ; Garbage?
0124    3806                    jrc     setup       ; Yes. Stick with current FCB
0126    21 005C                 lxi     H,DFCB      ; No. Get default FCB addr
0129    22 171E                 shld    FCBaddr     ; and make it our FCB addr.
                        ; Set up the 'worst byte' for being printed to console
012C                    setup:
012C    3A 0103                 lda     wrstbyt
012F    F5                      push    PSW         ; Split up the 'worst byte'
0130    07                      rlc                 ;into two chrs and save
0131    07                      rlc                 ;for 'Writing XXh fixed'
0132    07                      rlc                 ;in summary print-outs.
0133    07                      rlc
0134    CD 0144                 call    savnbl
0137    32 063E                 sta     wbchr1      ; 1st chr of wrstbyt
013A    F1                      pop     PSW
013B    CD 0144                 call    savnbl
013E    32 063F                 sta     wbchr2      ; 2nd chr of wrstbyt
0141    C3 0150                 jmp     BEGIN       ; preliminaries finished
                        ;----------
                        ;                       Subroutine: savnbl
                        ; Save a nibble
0144    E60F            savnbl: ani     0Fh
0146    FE0A                    cpi     10
0148    FA 014D                 jm      chr
014B    C607                    adi     07
014D    C630            chr:    adi     '0
014F    C9                      ret
```

```
                        ;--------------------
                        ; MAIN BODY - First - Initialize and obtain initial
                        ;                    instructions on which density and
                        ;                    which disks to test, and whether
                        ;                    or not to save the errors on disk
                        ;                    or network.
                        ;
0150    FB              BEGIN:  ei              ; enable interrupts
0151    3E00                    mvi     A,0     ; disable auto retries
0153    32 004E                 sta     BIOSrty ; in BIOS
0156    32 162B                 sta     dskflg  ; reset disk write flag
0159    CD 1825                 call    stoMAPbyt ; Save current disk select
015C    21 198F                 lxi     H,crlf  ; space down a line.
015F    CD 1385                 call    prtmsg
0162    21 059B                 lxi     H,initMSG ; print initial message
0165    CD 1385                 call    prtmsg

                        ; Find out if user wants buffer saved on disk.
0168    21 032B         dskask: lxi     H,dskmsg ; 'write to A:,B:,C:, or D:?'
016B    CD 1385                 call    prtmsg
016E    CD 1407                 call    CONIN
0171    F5                      push    PSW
0172    32 1391                 sta     retdsk  ; place chr in message
0175    21 1391                 lxi     H,retMSG
0178    CD 1385                 call    prtmsg  ; echo chr and space down
017B    F1                      pop     PSW
017C    FE59                    cpi     'Y'     ; write=yes?
017E    280D                    jrz     findED  ; Write=yes. Find error disk
0180    FE4E                    cpi     'N'     ; write=no?
0182    CA 0209                 jz      clrERR  ; Clear Error counts.
0185    21 0434                 lxi     H,errmsg ; bad entry. ask again
0188    CD 1385                 call    prtmsg
018B    18DB                    jmpr    dskask

                        ; User wants buffer saved on disk. Which diskname?
018D    21 0360         findED: lxi     H,CPMdskmsg  ; 'A,B,C, or D' msg
0190    CD 1385                 call    prtmsg
0193    CD 1353                 call    getCPMdsk
0196    32 1538                 sta     ERRDSK  ; disk for holding errs
0199    4F                      mov     C,A
019A    CD 1443                 call    SELDSK  ; select disk 1st thru BIOS
019D    CD 1825                 call    stoMAPbyt ; Save this MAP byte.  It
                                                ; will be used for dumping
                                                ; every 128 chrs (from the
                                                ; buffer) to this assignment.

                        ; Kill,create,open,write, and close the file.
01A0    21 0372                 lxi     H,stndbymsg
01A3    CD 1385                 call    prtmsg  ; Tell user to wait.
01A6    3A 1538                 lda     ERRDSK
01A9    5F                      mov     E,A
01AA    0E0E                    mvi     C,logdsk
01AC    CD 0005                 call    BDOS    ; select D disk thru BDOS
01AF    0E13                    mvi     C,kill
01B1    ED5B 171E               lied    FCBaddr ; File Control Block
```

```
01B5    CD 0005             call    BDOS        ; kill the file CONFILE.TXT
01B8    0E16                mvi     C,create
01BA    ED5B 171E           lded    FCBaddr
01BE    CD 0005             call    BDOS        ; create the file
01C1    0E0F                mvi     C,open
01C3    ED5B 171E           lded    FCBaddr
01C7    CD 0005             call    BDOS        ; attempt an open
01CA    0E1A                mvi     C,dma
01CC    11 045C             lxi     D,hdfile    ; file header addr
01CF    CD 0005             call    BDOS
01D2    0E15                mvi     C,wnr       ;'write next record'
01D4    ED5B 171E           lded    FCBaddr
01D8    CD 0005             call    BDOS        ; write a file header
01DB    21 0315             lxi     H,dskerr
01DE    FE00                cpi     0
01E0    F5                  push    PSW
01E1    C4 1385             cnz     prtmsg      ; print err msg
01E4    F1                  pop     PSW
01E5    C2 02D1             jnz     exit        ; die if cant write
01E8    0E10                mvi     C,close
01EA    ED5B 171E           lded    FCBaddr
01EE    CD 0005             call    BDOS        ; close the file
01F1    0E1A                mvi     C,dma
01F3    11 0080             lxi     D,80h
01F6    CD 0005             call    BDOS        ; reset the dma
01F9    21 162C             lxi     H,dskbuf
01FC    22 1628             shld    bufchr      ; begin disk buffer
01FF    3E00                mvi     A,0
0201    32 162A             sta     bufcnt      ; zero the counter
0204    3EFF                mvi     A,0FFh
0206    32 162B             sta     dskflg      ; set the disk write flag

                    ; Clear the error count locations.
0209    21 1489     clrERR: lxi     H,Terrors
020C    11 148A             lxi     D,Terrors+1
020F    01 000A             lxi     B,10
0212    3600                mvi     M,0
0214    EDB0                LDIR                ; Clear Track error locs

0216    21 149A             lxi     H,Serrors
0219    11 149B             lxi     D,Serrors+1
021C    01 000A             lxi     B,10
021F    3600                mvi     M,0
0221    EDB0                LDIR                ; Clear Sector error locs


                    ; Get the density.
0223    21 0390     finden:lxi      H,denmsg
0226    CD 1385             call    prtmsg      ; Ask user the density
0229    CD 1407             call    CONIN       ; fetch console chr
022C    F5                  push    PSW
022D    32 1391             sta     retdsk      ; place chr in message
0230    21 1391             lxi     H,retMSG
0233    CD 1385             call    prtmsg      ; echo chr and space down
0236    F1                  pop     PSW
```

```
0237  0600            mvi   B,SDbyt     ; Prepare for sing density
0239  FE53            cpi   'S'         ; Single density?
023B  280E            jrz   stodens     ; Yes. Save register B
023D  0640            mvi   B,DDbyt     ; No. Prepare for doub dens
023F  FE44            cpi   'D'         ; Double iensity?
0241  2808            jrz   stodens     ; Yes. Save register B
0243  21 0434         lxi   H,errmsg    ; No. Load error message
0246  CD 1385         call  prtmsg      ; Print error message
0249  18D8            jmpr  findden     ; and ask again.
024B  78       stodens:mov   A,B
024C  32 171C         sta   DENSbyt     ; Store flop dens code.
024F  21 0D00         lxi   H,26*128    ; set BUFLEN for sing dens
0252  2803            jrz   ..0         ; 0 is sing density
0254  21 1A00         lxi   H,26*256    ; set BUFLEN for doub dens
0257  22 1485    ..0:  shld  BUFLEN

                ; Get the test disks.
025A  21 03BB   findTD: lxi   H,tstmsg  ; 'Test Disk:' msg
025D  CD 1385         call  prtmsg
0260  21 14B5         lxi   H,TstUnits ; List of Units to test
0263  E5        ..1:  push  H           ; Save the list addr
0264  21 05CB         lxi   H,dskQST    ; ask which disk
0267  CD 1385         call  prtmsg
026A  CD 1333         call  getdsk      ; get a disk number to test
                                        ; FF returned = car ret
026D  E1              pop   H
026E  77              mov   M,A         ; Meanwhile load in disk num
026F  23              inx   H           ; And get to next list addr.
0270  FEFF            cpi   0FFh        ; More units?
0272  282A            jrz   ..3         ; No. All data collected.
0274  32 1483         sta   DSKNAME     ; store latest valid disk nam
0277  0602            mvi   B,2
0279  E5              push  H
027A  CD 1832         call  chaMAPbyt ; Select latest disk
027D  E1              pop   H
                ; Check for same disk selection for testing/errors
027E  3A 162B         lda   dskflg
0281  FEFF            cpi   0FFh        ; disk flag set?
0283  20DE            jrnz  ..1         ; NO. Bypass this test
0285  E5              push  H           ; Yes. Test for ambiguity.
0286  CD 1473         call  CPMmap      ; A = disks MAPbyte
0289  E6BF            ani   10111111b ; Turn off sing/doub dens bit
028B  47              mov   B,A
028C  3A 171D         lda   MAPbyt
028F  E6BF            ani   10111111b ; Turn off sing/doub dens bit
0291  B8              cmp   B           ; Compare that with our
0292  E1              pop   H
0293  20CE            jrnz  ..1         ; drives not same. Get next.
0295  21 03F1         lxi   H,drimsg  ; load 'Ambiguous case' msg
0298  CD 1385         call  prtmsg
029B  C3 0150         jmp   BEGIN     ; start over
                ;----------
                ;  ALL TEST DATA ENTERED AND CHECKED
                ;----------
029E  21 198F   ..3:  lxi   H,crlf
```

```
02A1    CD 1385             call    prtmsg    ; Space down
                    ; Build a linear table.
02A4    21 2000             lxi     H,lintab  ; build a linear table
02A7    01 4C01             lxi     B,76<3+1
02AA    CD 02F5             call    FILLTAB
02AD    21 204C             lxi     H,wrtab   ; build pseudo random table
02B0    01 4C15             lxi     B,76<3+21
02B3    CD 02F5             call    FILLTAB
02B6    21 2098             lxi     H,rdtab   ; build pseudo random table
02B9    01 4C1F             lxi     B,76<3+31
02BC    CD 02F5             call    FILLTAB
                    ; Print a command summary.
02BF    CD 1371             call    waitcr    ; wait for a cr
02C2    21 156D             lxi     H,HELPmsg
02C5    CD 1385             call    prtmsg    ; print command summary
02C8    21 198F             lxi     H,crlf
02CB    CD 1385             call    prtmsg    ; and space down a line
                    ; Start the test.
02CE    CD 04DD             call    TEST
```

```
                        ;*****************************************
                        ;*   PRE-TEST messages and subroutines   *
                        ;*****************************************

02D1    21 0305         exit:   lxi     H,exitMSG ; print exit message
02D4    CD 1385                 call    prtmsg
02D7    CD 1841                 call    resMAPbyt ; Restore original disk.
02DA    C3 0000                 jmp     0         ; WBOOT

02DD                    errexit:
02DD    E5                      push    H         ; save err msg address
02DE    3E00                    mvi     A,0
02E0    32 1537                 sta     errflg    ; turn off error flag
02E3    CD 1385                 call    prtmsg
02E6    CD 13FF                 call    CONST     ; console chr ready?
02E9    E1                      pop     H
02EA    FEFF                    cpi     0FFh
02EC    20EF                    jrnz    errexit   ; loop til a chr is ready
02EE    E5                      push    H
02EF    CD 16DC                 call    prtbuf    ; print err summary
02F2    E1                      pop     H
02F3    18E8                    jmpr    errexit   ; loop back forever

                        ;----------
                        ; Fill a table (b=length, c=increment)
02F5                    FILLTAB:
02F5    3E01                    mvi     A,1       ; start at 1
02F7    50                      mov     D,B       ; save length
02F8    77              ..1:    mov     M,A       ; put into memory
02F9    81                      add     C         ; add the increment
02FA    BA                      cmp     D         ; test for overflow
02FB    FA 0301                 jm      ..2
02FE    2801                    jrz     ..2
0300    92                      sub     D         ; if so, get back into range
0301    23              ..2:    inx     H         ; update HL
0302    10F4                    djnz    ..1       ; loop until done
0304    C9                      ret

0305    0D0A45584954 exitMSG:.asciz  [cr][lf]'EXIT FDTEST'[cr][lf]
0315    0D0A43616E6E dskerr: .asciz  [cr][lf]'Cannot access disk.'
032B    0D0A53617665 dskmsg: .ascii  [cr][lf]'Save ERRORS and loop info on '
034A    6469736B206F         .asciz  'disk or HiNet? (Y/N) '
0360                    CPMdskmsg:
0360    28412C20422C         .asciz  '(A, B, C, or D) ?'
0372                    stndbymsg:
0372    0D0A57616974         .asciz  [cr][lf]'Wait - Creating Buffer File'
0390    0D0A54657374 denmsg: .ascii  [cr][lf]'Test in Single or Double '
03AB    64656E736974         .asciz  'density? (S/D) '
03BB    0D0A54657374 tstmsg: .ascii  [cr][lf]'Test which drives? (Carria'
03D7    676520726574         .asciz  'ge return to terminate)'[cr][lf]
03F1    0D0A416D6269 drimsg: .ascii  [cr][lf]'Ambiguous drive selections. '
040F    2043616E6E6F         .ascii  ' Cannot save buffer on test diskette.'
0434    0D0A496E636F errmsg: .ascii  [cr][lf]'Incorrect Entry. '
0447    20506C656173         .asciz  ' Please try again.'[cr][lf]
                        ; Header for the CP/M file which will hold errors.
```

```
045C    0D0A464C4F50 hdfile: .ascii  [cr][lf]'FLOPPY DISK SUMMARY FILE'
0476    202020202020        .ascii
0496    0D0A2A2A2A2A        .ascii  [cr][lf]'****** **** ******* ****'
04B0    202020202020        .ascii
04D0    202020202020        .asciz  '        '[cr][lf]'  '
```

```
                        ;----------
                        ; TEST the disk
04DD                    TEST:
04DD    3E00                    mvi     A,0
04DF    32 14AD                 sta     fixflg  ; flag: fixed pattern off
04E2    3E01                    mvi     A,1     ; start test at loop 1
04E4    32 1487         tstloop:sta     LOOPS   ; store it away
04E7    07                      rlc
04E8    32 1488                 sta     RANSEED ; calculate first seed
04EB    3EFF                    mvi     A,0FFh
04ED    32 1537                 sta     errflg  ; turn on for 'LOOP' msg
04F0    21 05E0                 lxi     H,loopMSG ; print the loop message
04F3    CD 1385                 call    prtmsg
04F6    3A 1487                 lda     LOOPS
04F9    CD 13A2                 call    prtA
04FC    21 1988                 lxi     H,space5
04FF    CD 1385                 call    prtmsg
0502    CD 1850                 call    prttime ; print the time & date
0505    3E00                    mvi     A,0
0507    32 1537                 sta     errflg  ; turn buffer write flag off
050A    CD 170F                 call    putEOF

                        ;Write and test in a linearly-written pattern
050D    21 05E6         tststrt:lxi     H,wrlnMSG ; first write linearly
0510    22 14AF                 shld    dskfunc ; save current function
0513    CD 1385                 call    prtmsg
0516    21 2000                 lxi     H,lintab
0519    CD 064B                 call    wrtest
051C    21 05FA                 lxi     H,rdlnMSG ; then read linearly
051F    22 14AF                 shld    dskfunc ; save current function
0522    CD 1385                 call    prtmsg
0525    21 2000                 lxi     H,lintab
0528    CD 069F                 call    rdtest
052B    21 0622                 lxi     H,rdrnMSG ; then read randomly
052E    22 14AF                 shld    dskfunc ; save current function
0531    CD 1385                 call    prtmsg
0534    21 2098                 lxi     H,rdtab
0537    CD 069F                 call    rdtest

053A    3A 1488                 lda     RANSEED ; change the seed
053D    3C                      inr     A
053E    32 1488                 sta     RANSEED

                        ;Write and test a randomly-written pattern
0541    21 060E                 lxi     H,wrrnMSG ; next, write randomly
0544    22 14AF                 shld    dskfunc ; save current function
0547    CD 1385                 call    prtmsg
054A    21 204C                 lxi     H,wrtab
054D    CD 064B                 call    wrtest
0550    21 0622                 lxi     H,rdrnMSG ; then read randomly
0553    22 14AF                 shld    dskfunc ; save the current function

0556    CD 1385                 call    prtmsg
0559    21 2098                 lxi     H,rdtab
055C    CD 069F                 call    rdtest
```

```
055F    21 05FA             lxi     H,rdlnMSG ; then read linearly
0562    22 14AF             shld    dskfunc   ; save current function
0565    CD 1385             call    prtmsg
0568    21 2000             lxi     H,lintab
056B    CD 069F             call    rdtest

                    ;Write and test the worst byte in a fixed pattern
056E    32 14AD             sta     fixflg  ; set fixed pattern flag
0571    21 0636             lxi     H,wrwrstMSG ; write worst byte linearly
0574    22 14AF             shld    dskfunc     ; save current function
0577    CD 1385             call    prtmsg
057A    21 2000             lxi     H,lintab
057D    CD 064B             call    wrtest

0580    21 05FA             lxi     H,rdlnMSG ; then read linearly
0583    22 14AF             shld    dskfunc   ; save current function
0586    CD 1385             call    prtmsg
0589    21 2000             lxi     H,lintab
058C    CD 069F             call    rdtest
058F    3E00                mvi     A,0
0591    32 14AD             sta     fixflg  ; turn off fixed pattern

0594    3A 1487             lda     LOOPS
0597    3C                  inr     A
0598    C3 04E4             jmp     tstloop   ; read and write again

059B    20466C6F7070 initMSG:.ascii  ' Floppy Disk Test '
05AD    322E                .byte   version+'0','.'
05AF    3033                .byte   revision/10+'0',revision%10+'0'
05B1    20666F722074        .ascii  ' for the DSC3 and DSC4'
05C7    0D0A0A00            .asciz  [cr][lf][lf]
05CB    4469736B204E dskQST: .asciz 'Disk Number (0-7) : '
05E0    4C4F4F502000 loopMSG:.asciz 'LOOP '
05E6    57726974696E wrlnMSG:.asciz 'Writing linearly.'[cr][lf]
05FA    52656164696E rdlnMSG:.asciz 'Reading linearly.'[cr][lf]
060E    57726974696E wrrnMSG:.asciz 'Writing randomly.'[cr][lf]
0622    52656164696E rdrnMSG:.asciz 'Reading randomly.'[cr][lf]
0636    57726974696E wrwrstMSG: .ascii 'Writing '
063E    30           wbchr1: .byte  '0'     ; worst byte chr 1
063F    30           wbchr2: .byte  '0'     ; worst byte chr 2
0640    682066697865        .asciz  'h fixed.'[cr][lf]
```

```
                        ;-----------
                        ; Subroutine:   wrtest
                        ;
                        ; This subroutine supervises the write procedure.
                        ; A list of units under test is stored at TstUnits.
                        ; The list is terminated by an FFh.
  064B                  wrtest:
  064B    22 14B1               shld    wrparameter     ; Save test parameter
                        ; First we check that indeed there are units to test
  064E    21 14B5               lxi     H,TsTUnits      ; Get addr of test list
  0651    0600                  mvi     B,0             ; Zero out register B
  0653    7E            ..test: mov     A,M
  0654    23                    inx     H               ; Get to next list addr
  0655    FEFF                  cpi     0FFh            ; End of list?
  0657    2806                  jrz     ..check         ; Yes.
  0659    FE80                  cpi     80h             ; Is this Logged out?
  065B    28F6                  jrz     ..test          ; Yes. Keep looking
  065D    06FF                  mvi     B,0FFh          ; No. Set flag in B
  065F    78            ..check:mov     A,B             ; Check for flag
  0660    FEFF                  cpi     0FFh            ; Is it set?
  0662    21 0BB8               lxi     H,TermLOG       ; Get err msg ready
  0665    C2 02DD               jnz     errexit         ; No. Die.

                        ; At least one unit is logged in. Proceed with test.
  0668    21 14B5               lxi     H,TstUnits
  066B    7E            ..wr1:  mov     A,M             ; Get unit num to test
  066C    23                    inx     H               ; and get next list addr

  066D    FEFF                  cpi     0FFh            ; Is this last unit?
  066F    C8                    rz                      ; Yes.  Return.
  0670    FE80                  cpi     80h             ; Is unit logged out?
  0672    28F7                  jrz     ..wr1           ; Yes. Get the next one
  0674    E5                    push    H               ; No. Save unit addr
  0675    32 1483               sta     DSKNAME         ; No. Save unit num
  0678    0602                  mvi     B,2             ; Write thru CP/M 'C'
  067A    CD 1832               call    chaMAPbyt       ; Select that unit
  067D    2A 14B1               lhld    wrparameter     ; Restore test paramtr
  0680    CD 0686               call    wr1unit         ; Write to that unit,
  0683    E1                    pop     H               ; Restore unit address
  0684    18E5                  jmpr    ..wr1           ; and write again.
                        ;----------
                        ; Write tracks from table
  0686                  wr1unit:
  0686    E5                    push    H       ; save HL
  0687    CD 143B               call    HOME    ; home the disk
  068A    E1                    pop     H       ; restore HL
  068B    064C                  mvi     B,76    ; table has 76 entries
  068D    C5            ..1:    push    B       ; save BC and HL
  068E    E5                    push    H
  068F    7E                    mov     A,M     ; get track from table
  0690    32 1484               sta     TRACK   ; store in track
  0693    CD 0D20               call    wrtrk   ; write it
  0696    C4 0709               cnz     wrerr   ; check for errors
  0699    E1                    pop     H       ; restore BC and HL
  069A    C1                    pop     B
```

```
069B    23              inx     H       ; update HL
069C    10EF            djnz    ..1     ; repeat until no more tracks
069E    C9              ret

                ;-----------
                ; Subroutine:   rdtest
                ;
                ; This subroutine supervises the read procedure and
                ; reads from each unit under test.
                ; A list of units under test is stored at TstUnits.
                ; The list is terminated by an FFh.
069F            rdtest:
069F    22 14B3         shld    rdparameter     ; Save test parameter
                ; First we check that indeed there are units to test
06A2    21 14B5         lxi     H,TsTUnits      ; Get addr of test list
06A5    0600            mvi     B,0             ; Zero out register B
06A7    7E      ..test: mov     A,M
06A8    23              inx     H               ; Get to next list addr
06A9    FEFF            cpi     0FFh            ; End of list?
06AB    2806            jrz     ..check         ; Yes.
06AD    FE80            cpi     80h             ; Is this Logged out?
06AF    28F6            jrz     ..test          ; Yes. Keep looking
06B1    06FF            mvi     B,0FFh          ; No. Set flag in B
06B3    78      ..check:mov     A,B             ; Check for flag
06B4    FEFF            cpi     0FFh            ; Is it set?
06B6    21 0BB8         lxi     H,TermLOG       ; Get err msg ready
06B9    C2 02DD         jnz     errexit         ; No. Die.
                ; At least one unit is logged in. Proceed with test.
06BC    21 14B5         lxi     H,TstUnits
06BF    7E      ..rd1:  mov     A,M             ; Get unit num to test
06C0    23              inx     H               ; Get to next list addr
06C1    FEFF            cpi     0FFh            ; Is this last unit?
06C3    C8              rz                      ; Yes.  Return.
06C4    FE80            cpi     80h             ; Is unit logged out?
06C6    28F7            jrz     ..rd1           ; Yes. Get another one.
06C8    E5              push    H               ; No. Save unit addr
06C9    32 1483         sta     DSKNAME         ; No. Save unit num
06CC    0602            mvi     B,2             ; Write thru CP/M 'C'
06CE    CD 1832         call    chaMAPbyt       ; Select that unit
06D1    2A 14B3         lhld    rdparameter     ; Restore test paramtr
06D4    CD 06DA         call    rd1unit         ; Write to that unit
06D7    E1              pop     H               ; Restore list addr
06D8    18E5            jmpr    ..rd1           ; and write again.

                ;----------
                ; Read tracks from table and test.
06DA            rd1unit:
06DA    E5              push    H       ; home the disk
06DB    CD 143B         call    HOME
06DE    E1              pop     H
06DF    064C            mvi     B,76    ; the table has 76 entries
06E1    C5      ..1:    push    B       ; save BC and HL
06E2    E5              push    H
06E3    7E              mov     A,M     ; store table number from table
06E4    32 1484         sta     TRACK
```

```
06E7    21 3000             lxi     H,wrbuffer ; fill the wrbuffer for
06EA    ED4B 1485           lbcd    BUFLEN  ; compares
06EE    CD 0DA7             call    FILLPAT
06F1    CD 0D5A             call    rdtrk   ; read 3 times and check for
06F4    200A                jrnz    ..2     ; errors
06F6    CD 0D5A             call    rdtrk
06F9    2005                jrnz    ..2
06FB    CD 0D5A             call    rdtrk
06FE    2803                jrz     ..3
0700    CD 0737     ..2:    call    rderr   ; error has occured
0703    E1          ..3:    pop     H       ; restore BC and HL
0704    C1                  pop     B
0705    23                  inx     H       ; update HL
0706    10D9                djnz    ..1     ; repeat until no more tracks
0708    C9                  ret
```

```
                        ; A non-zero accumulator has been returned from wrtrk
                        ; or ritrk.  This number is an offset into the BIOS.
                        ; We use it as an offset into our error table. Error
                        ; flages are turned on, our current function (read or
                        ; write) is saved in ERRtry,and we read in and save the
                        ; NEC765 Error buffer (saved in the BIOS) and print
                        ; it out.

                        ;----------
                        ; Write and Read Error Routines
0709                    wrerr:
0709  F5                       push    PSW     ; save error code
070A  21 0D20                  lxi     H,wrtrk ; set up retry address
070D  22 14AB                  shld    ERRtry
0710  3EFF                     mvi     A,0FFh  ; turn on err flag
0712  32 1537                  sta     errflg
0715  21 078A                  lxi     H,wrerMSG ; print error message
0718  CD 1385                  call    prtmsg
071B  3A 1483                  lda     DSKNAME
071E  C630                     adi     '0'
0720  32 0C20                  sta     sumAnm
0723  21 0C1A                  lxi     H,sumA
0726  CD 1385                  call    prtmsg  ; print 'DISK' and disk number
0729  21 0C22                  lxi     H,sumAtrk
072C  CD 1385                  call    prtmsg
072F  3A 1484                  lda     TRACK   ; print track number
0732  CD 13A2                  call    prtA
0735  182C                     jmpr    errrtn  ; jump to common section

0737                    rderr:
0737  F5                       push    PSW     ; save error code
0738  21 0D5A                  lxi     H,ritrk ; set up retry address
073B  22 14AB                  shld    ERRtry
073E  3EFF                     mvi     A,0FFh  ; turn on err flag
0740  32 1537                  sta     errflg
0743  21 079A                  lxi     H,rderMSG ; print error message
0746  CD 1385                  call    prtmsg
0749  3A 1483                  lda     DSKNAME
074C  C630                     adi     '0'
074E  32 0C20                  sta     sumAnm
0751  21 0C1A                  lxi     H,sumA
0754  CD 1385                  call    prtmsg  ; print 'DISK' and disk number
0757  21 0C22                  lxi     H,sumAtrk
075A  CD 1385                  call    prtmsg
075D  3A 1484                  lda     TRACK   ; print track number
0760  CD 13A2                  call    prtA

                        ; Print out the current function (Writing linearly,
                        ; Reading randomly, etc)
                        ; Print out the NEC765 Flop Controller Chip
                        ; 16 chr buffer (kept in BIOS).
                        ; Restore error offset value (returned from
                        ; BIOS during Read/Write) and use it to jump
                        ; to appropriate location in the error table.
0763                    errrtn:
```

```
0763    21 1988             lxi     H,space5
0766    CD 1385             call    prtmsg      ; space over 5
0769    CD 1850             call    prttime     ; and print the time

076C    2A 0001             lhld    WBOOT
076F    11 009D             lxi     D,0A0h-3
0772    19                  dad     D           ; HL = Bios command buf addr
0773    11 153B             lxi     D,BIOSbuf   ; DE = Our buf holder
0776    01 0010             lxi     B,16        ; BC = num of bytes to copy
0779    EDB0                LDIR                ; Load in NEC765 err buf
077B    21 153B             lxi     H,BIOSbuf   ; into our buffer.
077E    CD 13BF             call    prt16       ; Print 16 byte NEC buffer
0781    F1                  pop     PSW         ; restore error offset numb

0782    21 07A6             lxi     H,errtab-3  ; jump to appropriate
0785    5F                  mov     E,A         ; routine
0786    1600                mvi     D,0
0788    19                  dad     D
0789    E9                  pchl

078A    0D0A20575249 wrerMSG:.asciz [cr][lf]' WRITE ERROR '
079A    0D0A20524541 rderMSG:.asciz [cr][lf]' READ ERROR '

07A9                errtab:
07A9    C3 07D0             jmp     SEEKerr
07AC    C3 07EE             jmp     SYNCerr
07AF    C3 081A             jmp     DATAerr
07B2    C3 0830             jmp     TRACerr
07B5    C3 085D             jmp     ENDTerr
07B8    C3 088D             jmp     IDerr
07BB    C3 08A1             jmp     ORUNerr
07BE    C3 08D2             jmp     SECTerr
07C1    C3 08F9             jmp     PROTerr
07C4    C3 092F             jmp     DENSerr
07C7    C3 096B             jmp     MADRerr
07CA    C3 0999             jmp     IOerr
07CD    C3 09B6             jmp     COMPerr

07D0                SEEKerr:
07D0    21 07D6             lxi     H,seekMSG ; Track error
07D3    C3 09CC             jmp     Ttypeerr
07D6    4552524F523A seekMSG:.asciz 'ERROR: equipment check '

07EE                SYNCerr:
07EE    21 07F4             lxi     H,syncMSG ; Chip and BIOS out of sync
07F1    C3 02DD             jmp     errexit   ; Terminal error
07F4    53594E432065 syncMSG:.asciz 'SYNC error - CPU and FDC out of sync '

081A                DATAerr:
081A    21 0820             lxi     H,dataMSG ; Sector type error
081D    C3 0A06             jmp     Stypeerr
0820    444154412043 dataMSG:.asciz 'DATA CRC error '

0830                TRACerr:
0830    21 0836             lxi     H,tracMSG ; Track type error
```

```
0833    C3 09CC             jmp     Ttypeerr
0836    545241432065 tracMSG:.asciz 'TRAC error - head over wrong cylinder '

085D                ENDTerr:
085D    21 0863             lxi     H,endtMSG ; DMA type error
0860    C3 0A3F             jmp     Dtypeerr
0863    454E44542065 endtMSG:.asciz 'ENDT error - read beyond end of cylinde
                    r '

088D                IDerr:
088D    21 0893             lxi     H,idMSG ; Sector type error
0890    C3 0A06             jmp     Stypeerr
0893    494420435243 idMSG: .asciz  'ID CRC error '

08A1                ORUNerr:
08A1    21 08A7             lxi     H,orunMSG ; DMA type error
08A4    C3 0A3F             jmp     Dtypeerr
08A7    444D41204F52 orunMSG:.asciz 'DMA ORUN - controller not seviced in ti
                    me '

08D2                SECTerr:
08D2    21 08D8             lxi     H,sectMSG ; Sector type error
08D5    C3 0A06             jmp     Stypeerr
08D8    534543542065 sectMSG:.asciz 'SECT error - Cannot find sector '

08F9                PROTerr:
08F9    21 0902             lxi     H,protMSG ; Write protected
08FC    CD 1385             call    prtmsg  ; print a message
08FF    C3 0105             jmp     start   ; and restart
0902    50524F542065 protMSG:.asciz 'PROT error - disk write-protected. REST
                    ART'[cr][lf]

092F                DENSerr:
092F    21 0935             lxi     H,densMSG ; Track type error
0932    C3 09CC             jmp     Ttypeerr
0935    44454E532065 densMSG:.asciz 'DENS error - wrong density (missing ID
                    address mark) '

096B                MADRerr:
096B    21 0971             lxi     H,madrMSG ; Track type error
096E    C3 09CC             jmp     Ttypeerr
0971    4D4144522065 madrMSG:.asciz 'MADR error - missing data address mark
                    '

0999                IOerr:
0999    21 099F             lxi     H,ioMSG ; Undefined error assume
099C    C3 09CC             jmp     Ttypeerr ; Track type error
099F    494E44455445 ioMSG: .asciz  'INDETERMINABLE error: '

09B6                COMPerr:
09B6    21 09BC             lxi     H,compMSG ; Sector type error
09B9    C3 0A06             jmp     Stypeerr
09BC    434F4D504152 compMSG:.asciz 'COMPARE error: '
```

```
                        ;----------
                        ; TRACK type error
09CC                    Ttypeerr:
09CC    CD 1385                 call    prtmsg      ; print callers message
09CF    21 1489                 lxi     H, Terrors ;
09D2    CD 0AAD                 call    incerr      ; update Track errors
09D5    01 0A00                 lxi     B,retry<8 ; retry the I/O
09D8    C5              ..1:    push    B
09D9    CD 143B                 call    HOME    ; rehome then retry
09DC    CD 0B0B                 call    rdwrtrk
09DF    C1                      pop     B
09E0    CA 09E4                 jz      ..2     ; retry a fixed number of times
09E3    0C                      inr     C
09E4    10F2            ..2:    djnz    ..1     ; and see how many were ok
09E6    79                      mov     A,C     ; save the answer
09E7    FE0A                    cpi     retry
09E9    CA 0A41                 jz      badtrack ; if none then bad track
09EC    CD 0CE2                 call    ratioc   ; otherwise print ratio
09EF    F5                      push    PSW      ; Save OK retry attempts
09F0    21 0C06                 lxi     H,CUR2msg  ; 'Current function: ' msg
09F3    CD 1385                 call    prtmsg
09F6    2A 14AF                 lhld    dskfunc    ; The cur func msg addr
09F9    CD 1385                 call    prtmsg
09FC    3E00                    mvi     A,0
09FE    32 1537                 sta     errflg  ; turn off the err flag
0A01    CD 170F                 call    putEOF
0A04    F1                      pop     PSW     ; Restore OK retry attempts
0A05    C9                      ret             ; and return

0A06                    Stypeerr:
0A06    CD 1385                 call    prtmsg    ; print callers message
0A09    CD 143B                 call    HOME      ; rehome the disk
0A0C    21 149A                 lxi     H,Serrors ; update Terrors
0A0F    CD 0AAD                 call    incerr
0A12    01 0A00                 lxi     B,retry<8 ; retry the I/O
0A15    C5              ..1:    push    B
0A16    CD 0B0B                 call    rdwrtrk ; just retry
0A19    C1                      pop     B
0A1A    2801                    jrz     ..2     ; retry a fixed number of times
0A1C    0C                      inr     C
0A1D    10F6            ..2:    djnz    ..1     ; and see how many were ok
0A1F    79                      mov     A,C     ; save the answer
0A20    FE0A                    cpi     retry
0A22    CA 0A41                 jz      badtrack ; if none then bad track
0A25    CD 0CE2                 call    ratioc   ; otherwise print ratio
0A28    F5                      push    PSW      ; Save OK retry attempts
0A29    21 0C06                 lxi     H,CUR2msg  ; 'Current function: ' msg
0A2C    CD 1385                 call    prtmsg
0A2F    2A 14AF                 lhld    dskfunc    ; The cur func msg addr
0A32    CD 1385                 call    prtmsg
0A35    3E00                    mvi     A,0     ; turn off the err flag
0A37    32 1537                 sta     errflg
0A3A    CD 170F                 call    putEOF
0A3D    F1                      pop     PSW     ; restore OK retry attempts
0A3E    C9                      ret             ; and return
```

```
0A3F                    Dtypeerr:
0A3F    18C5                    jmpr    Stypeerr ; just call is a Sector error

0A41                    badtrack:
0A41    21 0A73                 lxi     H,badtMSG ; print bad track message
0A44    CD 1385                 call    prtmsg
0A47    3A 1484                 lda     TRACK
0A4A    CD 13A2                 call    prtA      ; print bad track number
0A4D    21 0A8D                 lxi     H,bad1MSG ; print next part of msg
0A50    CD 1385                 call    prtmsg
0A53    3E0A                    mvi     A,retry
0A55    CD 13A2                 call    prtA      ; print amount of retries

0A58    21 0A9A                 lxi     H,bad2MSG ; print last part of msg
0A5B    CD 1385                 call    prtmsg
0A5E    21 0C06                 lxi     H,CUR2msg  ; 'Current function: ' msg
0A61    CD 1385                 call    prtmsg
0A64    2A 14AF                 lhld    dskfunc    ; The cur func msg addr
0A67    CD 1385                 call    prtmsg
0A6A    3E00                    mvi     A,0
0A6C    32 1537                 sta     errflg    ; turn off error flag
0A6F    CD 170F                 call    putEOF    ; and mark end of buffer
0A72    C9                      ret

0A73    0D0A42414420 badtMSG:.asciz  [cr][lf]'BAD TRACK FOUND: TRACK '
0A8D    20464149454C45 bad1MSG:.asciz  ' FAILED ALL '
0A9A    205245545259 bad2MSG:.asciz  ' RETRY ATTEMPTS.'[cr][lf]

                        ;----------
                        ; Subroutine:   incerr
                        ; Regs in:      HL = addr of error list
                        ; Increment the error count and log out the
                        ; disk from the test list if it shows FF errors.
                        ; We use the disknumber (DSKNAME) as an offset
                        ; into the list.
0AAD                    incerr:
0AAD    3A 1483                 lda     DSKNAME   ; get current disk
0AB0    1600                    mvi     D,0
0AB2    5F                      mov     E,A       ; DE = current disk number
0AB3    19                      dad     D         ; HL = Byte addr in error
                                                  ; list used by current disk.
0AB4    46                      mov     B,M       ; B = S or T errors on disk
0AB5    04                      inr     B         ; so far.
0AB6    3A 0104                 lda     maxerrs   ; Get max allowable errors
0AB9    B8                      cmp     B         ; max errs-current errs <= 0?
0ABA    3804                    jrc     ..1       ; Yes. Overflow. Log drv out.
0ABC    2802                    jrz     ..1       ; Yes. Overflow. Log drv out.
0ABE    34                      inr     M         ; No. Increment error count
0ABF    C9                      ret               ; and return.

                        ; Log out the present disk due to error overflow
0AC0                    ..1:
0AC0    21 0AE5                 lxi     H,oflowMSG
0AC3    CD 1385                 call    prtmsg      ; print overflow message
```

```
ØAC6   3EØØ              mvi     A,Ø          ; and turn off error flag
ØAC8   32 1537            sta     errflg
ØACB   CD 17ØF            call    putEOF
ØACE   21 14B4            lxi     H,TstUnits-1 ; get test list addr
ØAD1   3A 1483            lda     DSKNAME
ØAD4   47                 mov     B,A          ; get current disk name in B
ØAD5   23          ..loop: inx     H            ; get to next list addr
ØAD6   7E                 mov     A,M          ; get disk name from list
ØAD7   FEFF               cpi     ØFFh         ; end of list?
ØAD9   CA Ø5ØD            jz      tststrt      ; Yes. Restart main test loop
ØADC   9Ø                 sub     B            ; A = A - B
ØADD   FEØØ               cpi     Ø            ; Are list entry and dsk name
                                               ; identical?
ØADF   2ØF4               jrnz    ..loop       ; No. Keep looking for match
ØAE1   368Ø               mvi     M,8Øh        ; Yes Log out that disk
ØAE3   18FØ               jmpr    ..loop       ; and look for another match.

ØAE5                      oflowMSG:
ØAE5   ØDØA4552524F       .ascii  [cr][lf]'ERROR OVERFLOW - LOGGING'
ØAFF   204F55542044       .asciz  ' OUT DISK'[cr][lf]

ØBØB               rdwrtrk:
ØBØB   2A 14AB             lhld    ERRtry  ; load correct call address
ØBØE   E9                  pchl            ; and go there

                   ;----------
                   ; Subroutine :  prtdki
                   ; Regs  in:     none
                   ; Regs out:     none
                   ; Print the summary of all units under test.
ØBØF               prtdki:
ØBØF   3EØØ             mvi     A,Ø         ; Make error flag is
ØB11   32 1537           sta     errflg      ; off during summary.
ØB14   21 ØBDB           lxi     H,FDmsg     ; Print 'FDTEST' and
ØB17   CD 1385           call    prtmsg      ; space down a line.
ØB1A   CD 185Ø           call    prttime     ; print out the time & date
ØB1D   21 14B5           lxi     H,TstUnits
ØB2Ø   7E          ..prt1: mov     A,M         ; Get entry in list
ØB21   23                inx     H           ; Get to next addr
ØB22   FEFF              cpi     ØFFh        ; End of table?
ØB24   CA ØC4C           jz      prtcur      ; Yes. Print current function
ØB27   FE8Ø              cpi     8Øh         ; Is unit logged out?
ØB29   F5                push    PSW         ;----
ØB2A   E5                push    H
ØB2B   21 ØB7C           lxi     H,LOGoutMSG
ØB2E   CC 1385           cz      prtmsg      ; Print logged out message
ØB31   E1                pop     H           ; IF unit is logged out.
ØB32   F1                pop     PSW         ;----
ØB33   28EB              jrz     ..prt1      ; Yes. Skip it.
ØB35   32 1539           sta     tmpDSK      ; No. Valid unit name found.
ØB38   E5                push    H           ; Save table addr
ØB39   CD ØB3F           call    prt1unit    ; Print out that units info.
ØB3C   E1                pop     H
ØB3D   18E1              jmpr    ..prt1      ; Print next units info.
```

```
                        ; Print the summary for one unit
0B3F                    prt1unit:
0B3F    3A 1539                 lda     tmpDSK      ; print disk name
0B42    C630                    adi     '0'         ; make it ascii
0B44    32 0C20                 sta     sumAnm
0B47    21 0C1A                 lxi     H,sumA
0B4A    CD 1385                 call    prtmsg
0B4D    21 0C2C                 lxi     H,sumB
0B50    CD 1385                 call    prtmsg
0B53    21 149A                 lxi     H,Serrors
0B56    3A 1539                 lda     tmpDSK
0B59    1600                    mvi     D,0
0B5B    5F                      mov     E,A
0B5C    19                      dad     D
0B5D    7E                      mov     A,M         ; A = Sector errors on cur disk
0B5E    CD 13A2                 call    prtA        ; Print A
0B61    21 0C3B                 lxi     H,sumC
0B64    CD 1385                 call    prtmsg
0B67    21 1489                 lxi     H,Terrors
0B6A    3A 1539                 lda     tmpDSK
0B6D    1600                    mvi     D,0
0B6F    5F                      mov     E,A
0B70    19                      dad     D
0B71    7E                      mov     A,M         ; A = Track errors on cur disk
0B72    CD 13A2                 call    prtA        ; Print A
0B75    21 198F                 lxi     H,crlf
0B78    CD 1385                 call    prtmsg      ; Space down a line.
0B7B    C9                      ret

0B7C                    LOGoutMSG:
0B7C    2A2A20544845            .ascii  '** THE UNIT HERE WAS LOGGED '
0B98    4F5554204455            .asciz  'OUT DUE TO ERROR OVERFLOW. **'[cr][lf]
0BB8                    TermLOGout:
0BB8    2A2A20455252            .asciz  '** ERROR OVERFLOW ON ALL DRIVES **'
0BDB    0D0A20464454 FDmsg:     .ascii  [cr][lf]' FDTEST VER '
0BE9    322E                    .byte   version+'0','.'
0BEB    3033                    .byte   revision/10+'0',revision%10+'0'
0BED    2020202000              .asciz  '   '
0BF2    43757272656E CUR1msg:   .asciz  'Currently Testing: '
0C06    43757272656E CUR2msg:   .asciz  'Current Function : '
0C1A    4449534B3A20 sumA:      .ascii  'DISK: '
0C20    5800         sumAnm:    .asciz  'X'
0C22    2C2074726163 sumAtrk:   .asciz  ', track: '
0C2C    2C20522F5720 sumB:      .asciz  ', R/W ERRORS: '
0C3B    2C2054524143 sumC:      .asciz  ', TRACK ERRORS: '

                        ; Print the summary for the unit currently under test.
0C4C                    prtcur:
0C4C    21 0BF2                 lxi     H,CUR1msg
0C4F    CD 1385                 call    prtmsg      ; Print current info header
0C52    3A 1483                 lda     DSKNAME     ; print disk name
0C55    C630                    adi     '0'         ; make it ascii
0C57    32 0C20                 sta     sumAnm
0C5A    21 0C1A                 lxi     H,sumA
```

```
0C5D    CD 1385             call    prtmsg
0C60    21 0C22             lxi     H,sumAtrk
0C63    CD 1385             call    prtmsg
0C66    3A 1484             lda     TRACK
0C69    CD 13A2             call    prtA
0C6C    21 198F             lxi     H,crlf
0C6F    CD 1385             call    prtmsg  ; Space down.
0C72    21 0C06             lxi     H,CUR2msg
0C75    CD 1385             call    prtmsg
0C78    2A 14AF             lhld    dskfunc ; current disk function
0C7B    CD 1385             call    prtmsg
0C7E    21 198F             lxi     H,crlf
0C81    CD 1385             call    prtmsg  ; Space down a line.
0C84    C9                  ret             ; And finally return.

0C85                prtsum:
0C85    CD 13FF             call    CONST   ; has a char been read?
0C88    FEFF                cpi     0FFh    ; if not return
0C8A    C0                  rnz
0C8B    CD 1407             call    CONIN   ; else get it
0C8E    FE46                cpi     'F'     ; Fixed test wanted?
0C90    CA 1142             jz      FIXTEST ; Yes.
0C93    FE52                cpi     'R'     ; R? print the error buffer?
0C95    2004                jrnz    ..1
0C97    CD 16DC             call    prtbuf  ; if 'R' then print error buf
0C9A    C9                  ret
0C9B    FE53        ..1:    cpi     'S'     ; is it an 'S' (summary?)
0C9D    2004                jrnz    ..2
0C9F    CD 0B0F             call    prtdki  ; if 'S' then print disk status
0CA2    C9                  ret
0CA3    FE54        ..2:    cpi     'T'     ; If 'T' get new test disks
0CA5    CA 0209             jz      clrERR  ; Clear err counters and restrt
0CA8    FE51                cpi     'Q'     ; Is it a 'Q'? (quit)
0CAA    201B                jrnz    ..3     ; if not prt help msg & return
0CAC    21 0CCE             lxi     H,verMSG ; verify
0CAF    CD 1385             call    prtmsg
0CB2    CD 1407             call    CONIN
0CB5    32 1391             sta     retdsk
0CB8    21 1391             lxi     H,retMSG ; print answer
0CBB    CD 1385             call    prtmsg
0CBE    3A 1391             lda     retdsk  ; restore A
0CC1    FE59                cpi     'Y'     ; is answer yes?
0CC3    C0                  rnz
0CC4    C3 02D1             jmp     exit
0CC7                ..3:                    ; Bad entry. Print HELP message
0CC7    21 156D             lxi     H,HELPmsg
0CCA    CD 1385             call    prtmsg
0CCD    C9                  ret
0CCE    455849542046 verMSG: .asciz 'EXIT FDTEST (Y/N)? '

0CE2                ratioc:
0CE2    C5                  push    B       ; save C
0CE3    21 0D00             lxi     H,..A
0CE6    CD 1385             call    prtmsg  ; print ratio message
0CE9    C1                  pop     B       ; restore C
```

```
0CEA    79                      mov     A,C     ; into A
0CEB    CD 13A2                 call    prtA    ; and print it
0CEE    21 0D13                 lxi     H,..B   ; and print the rest of the
0CF1    CD 1385                 call    prtmsg  ; message
0CF4    3E0A                    mvi     A,retry
0CF6    CD 13A2                 call    prtA
0CF9    21 0D1C                 lxi     H,..C
0CFC    CD 1385                 call    prtmsg
0CFF    C9                      ret
0D00    0D0A52657472 ..A:       .asciz  [cr][lf]'Retry Failures: '
0D13    206F7574206F ..B:       .asciz  ' out of '
0D1C    2E0D0A00     ..C:       .asciz  '.'[cr][lf]

                    ;----------
                    ; Write a single track
0D20                wrtrk:
0D20    CD 0C85                 call    prtsum  ; print summary if requested
0D23    21 3000                 lxi     H,wrbuffer ; fill wrbuffer pattern
0D26    ED4B 1485               lbcd    BUFLEN
0D2A    CD 0DA7                 call    FILLPAT
0D2D    3A 1483                 lda     DSKNAME
0D30    0602                    mvi     B,2
0D32    CD 1832                 call    chaMAPbyt
0D35    3A 1484                 lda     TRACK
0D38    4F                      mov     C,A     ; first track
0D39    CD 144B                 call    SETTRK
0D3C    0E01                    mvi     C,1     ; then sector
0D3E    CD 1453                 call    SETSEC
0D41    01 3000                 lxi     B,wrbuffer
0D44    CD 145B                 call    SETDMA  ; then buffer address
0D47    ED4B 1485               lbcd    BUFLEN
0D4B    CD 147B                 call    SETBYT  ; and finally buffer length
0D4E    CD 146B                 call    WRITE   ; write the buffer
0D51    FE24                    cpi     36      ; check for errors
0D53    FA 0D58                 jm      ..0
0D56    3E24                    mvi     A,36    ; any out of range are ?
0D58    B7          ..0:        ora     A
0D59    C9                      ret

                    ;----------
                    ; Read a track and compare
0D5A                rdtrk:
0D5A    CD 0C85                 call    prtsum  ; print summary if requested
0D5D    3A 1483                 lda     DSKNAME
0D60    0602                    mvi     B,2
0D62    CD 1832                 call    chaMAPbyt
0D65    3A 1484                 lda     TRACK   ; set up bios calls
0D68    4F                      mov     C,A     ; first track
0D69    CD 144B                 call    SETTRK
0D6C    0E01                    mvi     C,1     ; then sector
0D6E    CD 1453                 call    SETSEC
0D71    01 5000                 lxi     B,rdbuffer ; then buffer address
0D74    CD 145B                 call    SETDMA
0D77    ED4B 1485               lbcd    BUFLEN  ; finally buffer length
0D7B    CD 147B                 call    SETBYT
```

```
ØD7E    CD 1463             call    READ    ; and read the track
ØD81    FE24                cpi     36      ; check for errors
ØD83    FA ØD88             jm      ..1
ØD86    3E24                mvi     A,36    ; any out of range are ?
ØD88    B7          ..1:    ora     A
ØD89    C0                  rnz
ØD8A    CD ØD8E             call    COMPARE ; and compare the data
ØD8D    C9                  ret

                    ;----------
                    ; Compare read and write buffers
ØD8E                COMPARE:
ØD8E    21 5000             lxi     H,rdbuffer
ØD91    11 3000             lxi     D,wrbuffer
ØD94    ED4B 1485           lbcd    BUFLEN
ØD98    1A          ..1:    ldax    D       ; get byte at (DE)
ØD99    BE                  cmp     M       ; compare with byte at (HL)
ØD9A    2008                jrnz    ..2     ; check for error !!!
ØD9C    23                  inx     H       ; update HL and DE
ØD9D    13                  inx     D
ØD9E    0B                  dcx     B       ; check for end of compare
ØD9F    79                  mov     A,C
ØDA0    B0                  ora     B
ØDA1    20F5                jrnz    ..1     ; loop back for more
ØDA3    C9                  ret
ØDA4    3E27        ..2:    mvi     A,39    ; compare error is 39 in table
ØDA6    C9                  ret

                    ;----------
                    ; Fill buffer with a pattern (random or fixed)
                    ; HL = Buffer address, BC = number of bytes to fill
ØDA7                FILLPAT:
ØDA7    3A 14AD             lda     fixflg
ØDAA    FEFF                cpi     ØFFh    ; fixed pattern desired?
ØDAC    CA ØDC9             jz      FIXFILL ; use second routine
ØDAF    3A 1488             lda     RANSEED ; seed with ranseed and track
ØDB2    07                  rlc
ØDB3    07                  rlc
ØDB4    07                  rlc
ØDB5    57                  mov     D,A
ØDB6    3A 1484             lda     TRACK
ØDB9    82                  add     D
ØDBA    77          ..1:    mov     M,A     ; write byte to memory
ØDBB    C62F                adi     2Fh     ; get next random #
ØDBD    EED1                xri     ØD1h
ØDBF    0F                  rrc
ØDC0    57                  mov     D,A     ; save random #
ØDC1    23                  inx     H       ; update HL
ØDC2    0B                  dcx     B       ; and BC
ØDC3    79                  mov     A,C     ; test for end of loop (BC=0)
ØDC4    B0                  ora     B
ØDC5    7A                  mov     A,D     ; restore random # to A
ØDC6    20F2                jrnz    ..1     ; and loop
ØDC8    C9                  ret
                    ;fill entire buffer with 'wrstbyt'
```

```
ØDC9                    FIXFILL:
ØDC9    3A Ø1Ø3                 lda     wrstbyt
ØDCC    77                      mov     M,A         ; put worst byte into buf
ØDCD    23                      inx     H           ; get to next buf addr
ØDCE    ØB                      dcx     B
ØDCF    79                      mov     A,C
ØDDØ    FEØØ                    cpi     Ø
ØDD2    2ØF5                    jrnz    fixfill     ; loop back
ØDD4    78                      mov     A,B
ØDD5    FEØØ                    cpi     Ø
ØDD7    2ØF0                    jrnz    fixfill     ; loop back
ØDD9    C9                      ret
```

```
                        ;*********************************************
                        ;*                                           *
                        ;*      FIXED TRACK TESTING SECTION          *
                        ;*                                           *
                        ;*********************************************

ØDDA    ØD0A46445445 fixmsg: .ascii  [cr][lf]'FDTEST fixed track '
ØDEF    526561642F57         .ascii  'Read/Write Option'[cr][lf][lf]
ØE03    205468697320         .ascii  ' This sub-program allows interact'
ØE24    69766520522F         .ascii  'ive R/W testing on a single track.'
ØE4E    2020416C6C0D         .ascii  '  All'[cr][lf]'Writes use only the'
ØE60    206669786564         .ascii  ' fixed pattern byte; '
ØE75    646973706C61         .ascii  "displayed during 'Writing XXh fixed'."
ØE9A    ØD0A41667465         .ascii  [cr][lf]'After each Read or Write is '
ØEB8    657865637574         .ascii  'executed, a hex number is printed to'
ØEDC    207468652063         .ascii  ' the console.'[cr][lf]
ØEEB    546865206E75         .ascii  'The number will be Ø if no errors are'
ØF10    206465746563         .ascii  ' detected, or it will be the offset '
ØF34    696E746F0D0A         .ascii  'into'[cr][lf]'the FDTEST error table.'
ØF51    2020496E2074         .ascii  '  In the event the number is not a Ø, '

ØF77    746865203136         .ascii  'the 16 byte NEC765'[cr][lf]'Floppy con'

ØF95    74726F6C6C65         .ascii  'troller error buffer will be'
ØFB1    20646973706C         .ascii  ' displayed.'
ØFBC    ØD0A0A00             .asciz  [cr][lf][lf]
ØFC0    ØD0A54686520 offmsg: .ascii  [cr][lf]'The buffer-to-disk write'
ØFDA    206973207465         .ascii  ' is temporarily disabled.'
ØFF3    ØD0A00               .asciz  [cr][lf]
ØFF6    ØD0A57697368 contmsg:.asciz  [cr][lf]'Wish to continue? (Y/N) '
1011    ØD0A0A       commsg: .ascii  [cr][lf][lf]
1014    2020436F6D6D         .ascii  '  Commands are:     R - Read and test'
1038    ØD0A                 .ascii  [cr][lf]
103A    202020202020         .ascii  '                    W - Write and test'
105F    ØD0A                 .ascii  [cr][lf]
1061    202020202020         .ascii  '                    S - Summary'
107F    ØD0A                 .ascii  [cr][lf]
1081    202020202020         .ascii  '                    T - Test a new '
10A3    747261636B0D         .ascii  'track'[cr][lf]
10AA    202020202020         .ascii  '                    D - Done, resume '
10CE    726567756C61         .ascii  'regular testing'[cr][lf]
10DF    202020202020         .ascii  '                    X - Done, eXit to'
1103    207379737465         .asciz  ' system.'[cr][lf][lf]
110F    204669786564 fxRDmsg:.asciz  ' Fixed Track Reading.'[cr][lf]
1127    204669786564 fxWRmsg:.asciz  ' Fixed Track Writing.'[cr][lf]

113F    ØØ           tmpflg: .byte   ØØ
1140    ØØ           fxtrack:.byte   ØØ
1141    ØØ           fxdisk: .byte   ØØ

                     ; Inform user he has selected fixed track testing.
1142                 FIXTEST:
1142    21 ØDDA              lxi     H,fixmsg
1145    CD 1385              call    prtmsg  ; Tell user he's in fixed mode
1148    21 ØFF6              lxi     H,contmsg
```

```
114B    CD 1385             call    prtmsg  ; Ask user if he wants to cont
114E    CD 1407     ..0:    call    CONIN   ; Get console chr
1151    F5                  push    PSW
1152    32 1391             sta     retdsk
1155    21 1391             lxi     H,retMSG
1158    CD 1385             call    prtmsg  ; Echo the chr
115B    F1                  pop     PSW
115C    FE4E                cpi     'N'     ; Chickened out?
115E    C8                  rz              ; Yes. Resume testing.
115F    FE59                cpi     'Y'     ; Brave?
1161    2808                jrz     fxstart ; Yes. Start initialization
1163    21 0434             lxi     H,errmsg
1166    CD 1385             call    prtmsg  ; Wrong entry. Print message
1169    18E3                jmpr    ..0     ; and ask again.

                    ; Initialize. Save users current disk-write option.
                    ; Disable disk-write option if enabled. Set fixed
                    ; pattern flag. (used in FILLPAT by wrtrk/rdtrk.
116B    3EFF        fxstart:mvi     A,0FFh
116D    32 14AD             sta     fixflg  ; Set fixed pattern write flag
1170    3A 162B             lda     dskflg  ; Save current disk flag
1173    32 113F             sta     tmpflg  ; and make sure its off.
1176    FE00                cpi     0
1178    2873                jrz     ..1     ; Its off. Get test info
117A    3E00                mvi     A,0
117C    32 162B             sta     dskflg
117F    21 0FC0             lxi     H,offmsg; Say we turned off disk write
1182    CD 1385             call    prtmsg
1185    1866                jmpr    ..1     ; get test info

1187                ..dskmsg:
1187    0D0A496E7075        .asciz  [cr][lf]'Input disk number to test :'
11A5                ..trkmsg:
11A5    0D0A496E7075        .ascii  [cr][lf]'Input track number'
11B9    20746F207465        .asciz  ' to test (in hex) and <CR> :'
11D6                ..rdymsg:
11D6    0D0A456E7465        .asciz  [cr][lf]'Enter first command '

                    ; Get disk and track number to test.
11ED    21 1187     ..1:    lxi     H,..dskmsg
11F0    CD 1385             call    prtmsg
11F3    CD 1333             call    getdsk
11F6    21 0434             lxi     H,errmsg
11F9    FEFF                cpi     0FFh    ; Just carriage return?
11FB    CC 1385             cz      prtmsg  ; Yes. Print error message
11FE    28ED                jrz     ..1     ; and ask again.
1200    32 1483             sta     DSKNAME ; No. Its valid. Store it.
1203    21 11A5     ..2:    lxi     H,..trkmsg
1206    CD 1385             call    prtmsg
1209    CD 1326             call    putBUF  ; Get ascii chrs in conbuf
                    ; A = length of users chr string.
120C    21 0434             lxi     H,errmsg; Get error message ready
120F    FE00                cpi     0       ; Just carriage return?
1211    CC 1385             cz      prtmsg  ; Yes. Print error message
1214    28ED                jrz     ..2     ; and ask again.
```

```
1216  CD 12F8            call   makHBYT ; No. Make chrs into a byte.
1219  32 1484            sta    TRACK   ; Store it as the track.
121C  21 1011            lxi    H,commsg
121F  CD 1385            call   prtmsg  ; Print command summary
1222  21 11D6            lxi    H,..rdymsg
1225  CD 1385            call   prtmsg  ; Tell user to input first
                                        ; command.
1228  CD 122D     ..wait: call   readCON ; Wait for the first command
122B  18FB               jmpr   ..wait

                  ;----------
                  ; All data collected for FIXTEST
                  ;----------

122D              readCON:
122D  CD 13FF            call   CONST   ; Check for console input
1230  FEFF               cpi    0FFh    ; Console chr ready?
1232  C0                 rnz            ; No. Resume to rding/wrting
1233  CD 1407            call   CONIN   ; Yes. Read the chr.
1236  F5                 push   PSW
1237  32 1391            sta    retdsk
123A  21 1391            lxi    H,retmsg
123D  CD 1385            call   prtmsg  ; Echo the character to console
1240  F1                 pop    PSW

1241  FE52               cpi    'R'     ; Read and test?
1243  2009               jrnz   ..Wchk  ; No. Check for W
1245  21 110F            lxi    H,fxRDmsg
1248  22 14AF            shld   dskfunc ; Put cur func in summary
124B  C3 1293            jmp    fxread

124E  FE57        ..Wchk: cpi    'W'     ; Write and test?
1250  2009               jrnz   ..Schk  ; No. Check for S
1252  21 1127            lxi    H,fxWRmsg
1255  22 14AF            shld   dskfunc ; Put cur func in summary
1258  C3 1279            jmp    fxwrite

125B  FE53        ..Schk: cpi    'S'     ; Summary wanted?
125D  2004               jrnz   ..Dchk  ; No. Check for D
125F  CD 0C4C            call   prtcur  ; Yes. Print it
1262  C9                 ret            ; and return.

1263  FE44        ..Dchk: cpi    'D'     ; Done?
1265  2849               jrz    fxreturn

1267  FE54               cpi    'T'     ; Test New track?
1269  CA 116B            jz     fxstart

126C  FE58               cpi    'X'     ; Abort test?
126E  283D               jrz    fxabort

                  ; Invalid chr entered.
1270  21 1011            lxi    H,commsg;Bad entry. Print
1273  CD 1385            call   prtmsg  ;help summary
1276  C3 122D            jmp    readCON ;and return.
```

```
                        ;----------
                        ; Fixed test operating routines
1279                    fxwrite:
1279    21 2000                 lxi     H,lintab
127C    CD 0D20                 call    wrtrk   ; 0 = AOK write
127F    F5                      push    PSW     ; Save error code
1280    CD 13A2                 call    prtA    ; Print 00 console
1283    0E20                    mvi     C,' '   ; and space over.
1285    CD 1421                 call    CONOUT
1288    F1                      pop     PSW
1289    FE00                    cpi     00      ; Did write go OK?
128B    C4 12E2                 cnz     fxprtbuf; No. Print NEC error buffer
128E    CD 122D                 call    readCON ; Check for new con command
1291    18E6                    jmpr    fxwrite ; Write again and again....

1293                    fxread:
1293    21 2000                 lxi     H,lintab
1296    CD 0D5A                 call    rdtrk
1299    F5                      push    PSW     ; Save error code
129A    CD 13A2                 call    prtA    ; Print 00 console
129D    0E20                    mvi     C,' '   ; and space over.
129F    CD 1421                 call    CONOUT
12A2    F1                      pop     PSW
12A3    FE00                    cpi     00      ; Did write go OK?
12A5    C4 12E2                 cnz     fxprtbuf; No. Print NEC error buffer
12A8    CD 122D                 call    readCON ; Check for new con command.
12AB    18E6                    jmpr    fxread  ; Read again and again....

12AD    C3 02D1         fxabort:jmp     exit

12B0                    fxreturn:
12B0    21 12C2                 lxi     H,..rtnMSG
12B3    CD 1385                 call    prtmsg  ; Tell user were leaving
12B6    3A 113F                 lda     tmpflg
12B9    32 162B                 sta     dskflg  ; Restore disk flag.
12BC    31 7000                 lxi     sp,stack; Re-initiate stack
12BF    C3 0209                 jmp     clrERR  ; Restart FDTEST

12C2                    ..rtnMSG:
12C2    0D0A0A                  .ascii  [cr][lf][lf]
12C5    52657475726E            .asciz  'Returning to FDTEST'[cr][lf][lf]
12DC    05              conbuf: .byte   5                       ; CPM buffer length
12DD    2020202020              .byte   20h,20h,20h,20h,20h ; CP/M buffer space

                        ;--------------
                        ; Fixtest Subroutines
                        ;

                        ;---------
                        ; Subroutine: fxprtbuf
                        ; Load the NEC765 Flop Cont Chip Error buffer
                        ; into memory and print it out.
12E2                    fxprtbuf:
12E2    2A 0001                 lhld    WBOOT
```

```
12E5    11 009D             lxi     D,0A0h-3
12E8    19                  dai     D           ; HL = Bios command buf addr
12E9    11 153B             lxi     D,BIOSbuf   ; DE = Our buf holder
12EC    01 0010             lxi     B,16        ; BC = num of bytes to copy
12EF    EDB0                LDIR                ; Load in NEC765 err buf
12F1    21 153B             lxi     H,BIOSbuf   ; into our buffer.
12F4    CD 13BF             call    prt16       ; Print 16 byte NEC buffer
12F7    C9                  ret

                    ;----------
                    ;                   Subroutine: makHBYT
                    ; Regs  in:        A=length of console buffer
                    ; Regs out:        A=hex byte made from console buffer
                    ;Destroyed:        HL,A,B,C
                    ;Make a hex byte from one or two buffer chrs
12F8                makHBYT:
12F8    FE01                cpi     1           ; just one chr?
12FA    0600                mvi     B,0
12FC    21 12DE             lxi     H,conbuf+2  ; load 1st chr addr
12FF    2816                jrz     ..a1        ; process 1st chr only
                    ;multiply first chr by 16 and save in B
1301    7E                  mov     A,M         ; get 1st chr
1302    23                  inx     H           ; get to next chr
1303    FE41                cpi     'A'
1305    3804                jrc     ..a2
1307    E6DF                ani     0DFh        ; make all letters upper case
1309    D607                sui     'A'-'9'-1
130B    D630        ..a2:   sui     '0'
130D    FE00                cpi     0
130F    2806                jrz     ..a1        ; skip mult if nibble is 0
1311    B7                  ora     A           ; reset carry flag
1312    07                  rlc
1313    07                  rlc
1314    07                  rlc
1315    07                  rlc
1316    47                  mov     B,A         ;save 16*firstnumber in B
1317    7E          ..a1:   mov     A,M         ; get 2nd chr
1318    FE41                cpi     'A'
131A    FA 1321             jm      ..a3
131D    E6DF                ani     0DFh        ; make all letters upper case
131F    D607                sui     'A'-'9'-1
1321    D630        ..a3:   sui     '0'
1323    B7                  ora     A           ; reset the flags
1324    80                  add     B           ; add the two chrs
1325    C9                  ret

                    ;----------
                    ;                   Subroutine: putBUF
                    ; Regs  in:        HL=address of message to print
                    ; Regs out:        A =number of chrs put in buffer
                    ;Destroyed:
                    ;Put console input in a buffer
1326                putBUF:
1326    0E0A                mvi     C,bufread
1328    11 12DC             lxi     D,conbuf
```

```
132B    CD 0005         call    BDOS        ; put console input in buffer
132E    21 12DD         lxi     H,conbuf+1
1331    7E              mov     A,M         ; put num of buffer chrs in A
1332    C9              ret
```

```
                        ;**************************************************
                        ;*                                                *
                        ;*   CONSOLE I/O SUBOUTINES AND BIOS PRIMITIVES   *
                        ;*                                                *
                        ;**************************************************
                        ;----------
                        ; Get disk name, return 0FFh if cr
1333                    getdsk:
1333    CD 1407                 call    CONIN   ; get a char
1336    FE0D                    cpi     cr      ; Carriage return?
1338    2816                    jrz     ..3     ; Yes. Handle it specially.
133A    FE38                    cpi     '8'     ; Is it legal?
133C    30F5                    jrnc    getdsk  ; No. Fetch next chr.
133E    FE30                    cpi     '0'     ; Is it legal?
1340    38F1                    jrc     getdsk  ; No. Fetch next chr.

1342    F5              ..2:    push    PSW     ; save the char
1343    32 1391                 sta     retdsk  ; echo char then crlf
1346    21 1391                 lxi     H,retMSG
1349    CD 1385                 call    prtmsg
134C    F1                      pop     PSW     ; restore PSW
134D    D630                    sui     '0'     ; get disk select number
134F    C9                      ret

1350    3EFF            ..3:    mvi     A,0FFh  ; Put flag into reg A
1352    C9                      ret

                        ;----------
                        ; Get CP/M disk name (A, B, C, or D) from user
                        ;
1353                    getCPMdsk:
1353    CD 1407                 call    CONIN           ; get console chr
1356    FE03                    cpi     3               ; Is chr a ctrl-C?
1358    CA 02D1                 jz      exit            ; Yes. Abort.
135B    FE45                    cpi     'D'+1           ; Is it too large?
135D    30F4                    jrnc    getCPMdsk       ; Yes. Fetch next chr
135F    FE41                    cpi     'A'             ; Is it too small?
1361    38F0                    jrc     getCPMdsk       ; Yes. Fetch next chr
1363    F5                      push    PSW             ; No. Echo valid chr
1364    32 1391                 sta     retdsk
1367    21 1391                 lxi     H,retMSG
136A    CD 1385                 call    prtmsg
136D    F1                      pop     PSW
136E    D641                    sui     'A'             ; Make it 0,1,2, etc
1370    C9                      ret                     ; and return.


                        ;----------
                        ; Wait for a cr
1371                    waitcr:
1371    21 1395                 lxi     H,waitMSG ; print a message
1374    CD 1385                 call    prtmsg
1377    CD 1407         ..1:    call    CONIN   ; get a char
137A    FE0D                    cpi     cr      ; loop if cr
137C    20F9                    jrnz    ..1
```

```
137E    21 1392             lxi     H,crlfMSG ; print crlf
1381    CD 1385             call    prtmsg
1384    C9                  ret

                    ;----------
                    ; Print a message addressed by HL.
                    ; Return on a null ( 00 ).  A null is automatically
                    ; tagged onto the end of any '.asciz' command.
1385                prtmsg:
1385    7E                  mov     A,M     ; get next char
1386    B7                  ora     A
1387    C8                  rz              ; return if zero
1388    E5                  push    H       ; save HL
1389    4F                  mov     C,A     ; output the char
138A    CD 1421             call    CONOUT
138D    E1                  pop     H       ; restore HL
138E    23                  inx     H       ; increment to next char
138F    18F4                jmpr    prtmsg  ; print it

1391                retMSG:
1391    41          retdsk: .byte   'A'
1392    0D0A00      crlfMSG:.asciz  [cr][lf]
1395    547970652052 waitMSG:.asciz 'Type RETURN '

                    ;----------
                    ; Print the A register as a hex number
13A2                prtA:
13A2    F5                  push    PSW     ; save A
13A3    07                  rlc             ; get upper nibble
13A4    07                  rlc
13A5    07                  rlc
13A6    07                  rlc
13A7    CD 13AF             call    prtnib  ; print it
13AA    F1                  pop     PSW     ; restore A
13AB    CD 13AF             call    prtnib  ; print lower nibble
13AE    C9                  ret
                    ;----------
                    ; Print a nibble
13AF                prtnib:
13AF    E60F                ani     0Fh     ; mask to get nibble
13B1    FE0A                cpi     10      ; convert to ASCII
13B3    FA 13B8             jm      ..1
13B6    C607                adi     7
13B8    C630        ..1:    adi     '0'
13BA    4F                  mov     C,A
13BB    CD 1421             call    CONOUT
13BE    C9                  ret
                    ;-----------
                    ; Subroutine: prt16
                    ; Print header and 16 chracters to the console.
                    ; Used for printing out NEC765 BIOS error buffer.
                    ; Regs  in:     HL = address of 1st chr
13BF                prt16:
13BF    E5                  push    H
13C0    21 13E5             lxi     H,Headmsg
```

```
13C3    CD 1385             call    prtmsg
13C6    E1                  pop     H
13C7    3E10                mvi     A,16
13C9    F5          ..cnt:  push    PSW
13CA    E5                  push    H
13CB    7E                  mov     A,M
13CC    CD 13A2             call    prtA
13CF    0E20                mvi     C,' '
13D1    CD 1421             call    CONOUT
13D4    E1                  pop     H
13D5    23                  inx     H
13D6    F1                  pop     PSW
13D7    3D                  dcr     A
13D8    FE00                cpi     0
13DA    2802                jrz     ..rest
13DC    18EB                jmpr    ..cnt
13DE    21 198F     ..rest: lxi     H,crlf
13E1    CD 1385             call    prtmsg
13E4    C9                  ret
13E5    4E4543373635 headmsg: .asciz 'NEC765 Error Flag Buffer '
                    ;----------
                    ; Call bios directly using WBOOT in low memory
0001                WBOOT   =       1

13FF                CONST:
13FF    2A 0001             lhld    WBOOT
1402    11 0003             lxi     D,03h
1405    19                  dad     D
1406    E9                  pchl

1407                CONIN:
1407    21 1413             lxi     H,..1   ; return to ..1
140A    E5                  push    H
140B    2A 0001             lhld    WBOOT
140E    11 0006             lxi     D,06h
1411    19                  dad     D
1412    E9                  pchl

1413    FE03        ..1:    cpi     3       ; 3 is Control-C
1415    CA 02D1             jz      exit    ; Wboot if Control-C
1418    FE61                cpi     'a'     ; make result UPPER CASE
141A    D8                  rc
141B    FE7B                cpi     'z'+1
141D    D0                  rnc
141E    D620                sui     'a'-'A'
1420    C9                  ret

1421                CONOUT:
                    ;The chr arrives in Reg C.
1421    3A 1537             lda     errflg  ; is this part of an err msg?
1424    FEFF                cpi     0FFh
1426    200B                jrnz    chrout  ; just output the chr
1428    CD 16CF             call    wrtbuf  ; put chr in error buffer (R)
142B    3A 162B             lda     dskflg  ; do we put this into disk buf?
142E    FEFF                cpi     0FFh
```

```
1430    CC 16AC             cz      stobyt  ; put it in disk buffer

1433    2A 0001     chrout: lhld    WBOOT
1436    11 0009             lxi     D,09h
1439    19                  dad     D
143A    E9                  pchl

143B                HOME:
143B    2A 0001             lhld    WBOOT
143E    11 0015             lxi     D,15h
1441    19                  dad     D
1442    E9                  pchl

1443                SELDSK:
1443    2A 0001             lhld    WBOOT
1446    11 0018             lxi     D,18h
1449    19                  dad     D
144A    E9                  pchl

144B                SETTRK:
144B    2A 0001             lhld    WBOOT
144E    11 001B             lxi     D,1Bh
1451    19                  dad     D
1452    E9                  pchl

1453                SETSEC:
1453    2A 0001             lhld    WBOOT
1456    11 001E             lxi     D,1Eh
1459    19                  dad     D
145A    E9                  pchl

145B                SETDMA:
145B    2A 0001             lhld    WBOOT
145E    11 0021             lxi     D,21h
1461    19                  dad     D
1462    E9                  pchl

1463                READ:
1463    2A 0001             lhld    WBOOT
1466    11 0024             lxi     D,24h
1469    19                  dad     D
146A    E9                  pchl

146B                WRITE:
146B    2A 0001             lhld    WBOOT
146E    11 0027             lxi     D,27h
1471    19                  dad     D
1472    E9                  pchl

1473                CPMMAP:
1473    2A 0001             lhld    WBOOT
1476    11 0060             lxi     D,60h
1479    19                  dad     D
147A    E9                  pchl
```

```
147B                    SETBYT:
147B    2A 0001                 lhld    WBOOT
147E    11 0066                 lxi     D,66h
1481    19                      dad     D
1482    E9                      pchl

1483    00              DSKNAME: .byte  00
1484    00              TRACK:   .byte  00
1485    0000            BUFLEN:  .word  00
1487    00              LOOPS:   .byte  00
1488    00              RANSEED: .byte  00
1489    000000000000    Terrors: .byte  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
149A    000000000000    Serrors: .byte  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
14AB    0000            ERRtry:  .word  0000
14AD    0000            fixflg:  .word  0000    ; flag: fixed or random pattern
14AF    0000            dskfunc: .word  0000    ; current R/W func under test

2000                    lintab  =       2000h
204C                    wrtab   =       lintab+76
2098                    rdtab   =       wrtab+76

3000                    wrbuffer=       3000h
5000                    rdbuffer=       5000h
7000                    stack   =       7000h

14B1    0000            wrparameter:    .word   0000h   ; Temp holder for read
14B3    0000            rdparameter:    .word   0000h   ; & write parameters.
14B5                    TstUnits:
14B5                            .blkb   128             ; Space for list of
                                                        ; units under test.
                                                        ; FFh terminates list.
```

```
                        ;***********************************************
                        ;*                                             *
                        ;*  ERROR BUFFER DATA, DEFS, AND SUBROUTINES   *
                        ;*                                             *
                        ;***********************************************

7000                    erbuffer=       7000h
001A                    EOF     =       1Ah
1535    0000            errbufadr: .word 00h    ; Current spot in err buffer.
1537    00              errflg: .byte   0       ; Write flag for error buffer.
1538    00              ERRDSK: .byte   0       ; Holder of error disk number.
1539    00              tmpDSK: .byte   0       ; Temp holder of a disk number.
153A    00              orgdriv:.byte   0       ; Drive to boot to when done.

                        ; The NEC765 Error buffer results are stored here.
153B    000000000000    BIOSbuf:.byte   0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0

                        ; Output messages
154E    0D0A2A2A2A54    truncmsg:.asciz [cr][lf]'***TRUNCATED TEST LISTING***'
155D    0D0A0A202020    HELPmsg:.ascii  [cr][lf][lf]'     Command Summary'
1584    0D0A20202020            .ascii  [cr][lf]'     ------- --------'
159A    0D0A2046202D            .ascii  [cr][lf]' F - Fixed track testing'
15B4    0D0A2052202D            .ascii  [cr][lf]' R - Read out the error buffer'
15D4    0D0A2053202D            .ascii  [cr][lf]' S - Print the current summary'
15F4    0D0A2054202D            .ascii  [cr][lf]' T - Restart the test '
160C    0D0A2051202D            .ascii  [cr][lf]' Q - Optional Halt/Stop'
1625    0D0A00                  .asciz  [cr][lf]

1628    0000            bufchr: .word   0       ; filled in at START
162A    00              bufcnt: .byte   0       ; updated by stobyt,Diskit
162B    00              dskflg: .byte   0
162C                    dskbuf: .blkb   128     ; chr buffer for disk outp


                        ;-----------------
                        ;               Subroutine: stobyt
                        ; Regs  in: A (a console input or output character)
                        ; Regs out: A (unchanged character)
                        ; Regs destroyed: none
                        ; Variables used:      bufchr  -next chr in buffer
                        ;                      bufcnt  -chr counter
16AC                    STOBYT:
16AC    F5                      push PSW
16AD    C5                      push B
16AE    D5                      push D
16AF    E5                      push H
16B0    2A 1628                 lhld    bufchr          ; get next chr addr
16B3    71                      mov     M,C             ; write it to buffer
16B4    23                      inx     H               ; increment buff addr
16B5    22 1628                 shld    bufchr          ; save next chr addr
                        ;increment counter and test for full buffer
16B8    3A 162A                 lda     bufcnt
16BB    3C                      inr     A
16BC    FE80                    cpi     128             ; 128 chrs ready?
16BE    3806                    jrc     ..notfull       ; No.
```

```
16C0    CD 1746             call    Diskit          ; Yes. Put it on disk
                                                    ; if disk flag is set.
16C3    C3 17E4             jmp     BUFreset        ; Reset, and return.
16C6                    ..notfull:
16C6    32 162A             sta     bufcnt          ; save chr counter
16C9    79                  mov     A,C             ; restore chr
16CA    E1                  pop H
16CB    D1                  pop D
16CC    C1                  pop B
16CD    F1                  pop PSW
16CE    C9                  ret

                        ;------------------
                        ;               Subroutine: wrtbuf
                        ; Regs  in:
                        ; Regs out:
                        ; Destroyed:
                        ;Write a character to the error buffer
16CF                    WRTBUF:
16CF    E5                  push    H
16D0    2A 1535             lhld    errbufadr
16D3    71                  mov     M,C
16D4    23                  inx     H
16D5    361A                mvi     M,EOF   ; tag on an EOF (1Ah)
16D7    22 1535             shld    errbufadr
16DA    E1                  pop     H
16DB    C9                  ret
                        ;------------------
                        ;               Subroutine: prtbuf
                        ; Regs  in:
                        ; Regs out:
                        ; Destroyed:
                        ;
                        ;Print out the error buffer to the console
16DC                    PRTBUF:
16DC    21 198F             lxi     H,crlf
16DF    CD 1385             call    prtmsg  ;space down
16E2    3E00                mvi     A,0
16E4    32 1537             sta     errflg  ;make sure err flag is off
16E7    21 7000             lxi     H,erbuffer ; load err buf addr
16EA    7E              outchr: mov     A,M     ;get chr in A
16EB    FE1A                cpi     1Ah     ;EOF?
16ED    C8                  rz
16EE    4F                  mov     C,A     ;get chr in C
16EF    23                  inx     H       ;get to next buf adr
16F0    E5                  push    H       ;save buffer address
16F1    CD 1421             call    CONOUT
16F4    CD 13FF             call    CONST
16F7    FEFF                cpi     0FFh    ;chr ready?
16F9    E1                  pop     H       ;restore buffer addr
16FA    C2 16EA             jnz     outchr  ;if not, out next chr
16FD    E5                  push    H       ;save buffer addr
16FE    CD 1407             call    CONIN   ;eat the chr
1701    CD 13FF         ..1:    call    CONST   ;ready to resume?
1704    FEFF                cpi     0FFh
```

```
1706    20F9            jrnz    ..1     ;wait for next chr
1708    CD 1407         call    CONIN   ;eat the chr
170B    E1              pop     H       ;save buffer addr
170C    C3 16EA         jmp     outchr  ;resume outputting buffer
                ;------------------
                ;               Subroutine: putEOF
                ; Regs  in:     none
                ; Regs out:     none
                ; Destroyed:    none
                ;Put an End of File marker (1Ah) at end of buffer.
170F            putEOF:
170F    E5              push    H
1710    2A 1535         lhld    errbufadr
1713    3E1A            mvi     A,EOF
1715    77              mov     M,A
1716    23              inx     H
1717    77              mov     M,A
1718    23              inx     H
1719    77              mov     M,A
171A    E1              pop     H
171B    C9              ret
```

```
                        ;*************************************************
                        ;*                                               *
                        ;* DISK WRITE (OF CONSOLE OUTPUT) DEFS AND SUBS  *
                        ;*                                               *
                        ;*************************************************

0005                    BDOS    =       05h
000A                    bufread =       0Ah
000D                    reset   =       0Dh
000E                    logdisk =       0Eh
000F                    open    =       0Fh
0010                    close   =       10h
0013                    kill    =       13h
0016                    create  =       16h
0015                    wnr     =       15h         ; write next record
001A                    dma     =       1Ah
0000                    SDbyt   =       00000000b ; SD flop device code
0040                    DDbyt   =       01000000b ; DD flop device code
171C    00              DENSbyt:.byte   00h         ; Density for test
171D    00              MAPbyt: .byte   00h         ; Users current MAPbyte
                        ;
                        ; File Control Block Address.  (Either below or 5Ch)
171E    0000            FCBaddr: .word  0000h

                        ;Internal File Control Block
                        ;(For CP/M file management)
                        ;
1720    00              FCB:    .byte   0
1721    434F4E46494C            .ascii  'CONFILE TXT'
172C    000000000000            .byte   0,0,0,0,0,0,0,0,0,0,0,0,0
1739    000000000000            .byte   0,0,0,0,0,0,0,0,0,0,0,0,0
                        ;
                        ;-----------------
                        ;               Subroutine: Diskit
                        ; Regs  in:     None
                        ; Regs out:     None
                        ; Regs destroyed:       None
                        ; Variables used:       bufchr,bufcnt
                        ;
                        ;The 128 chr buffer (at dskbuf) is full. CONFILE.TXT
                        ;is opened, and the next record to write,  the FCB
                        ;record count is found and loaded into the FCB
                        ;current record.  We write to the next record
                        ;(dumping from the 128 chr buffer), then close the
                        ;file. Next, we check the current error buffer
                        ;location.  Restart the error buffer if
                        ;it has passed B000h.
1746                    DISKIT:
                        ;make sure disk-write flag is set
1746    3A 162B                 lda     dskflg
1749    FEFF                    cpi     0FFh    ; Is flag set?
174B    C0                      rnz             ; No. Return to stobyt
                        ;reset the bytes-per-read value
174C    01 0080                 lxi     B,128   ; normal I/O byte value
174F    CD 147B                 call    SETBYT
```

```
                        ;select the error disk
1752    3A 1538                 lda     ERRDSK
1755    CD 1841                 call    resMAPbyt
                        ;reset the dma
1758    11 0080                 lxi     D,80h
175B    0E1A                    mvi     C,dma
175D    CD 0005                 call    BDOS
                        ;open the file
175G    0E0F                    mvi     C,open  ; prepare to open file
1762    ED5B 171E               lded    FCBaddr
1766    CD 0005                 call    BDOS
                        ;Get to last record of file and
                        ; find FCB's record count then set FCB's current record
1769    2A 171E                 lhld    FCBaddr
176C    11 000F                 lxi     D,15
176F    19                      dad     D
1770    7E                      mov     A,M     ; get rec count in acc
1771    2A 171E                 lhld    FCBaddr
1774    11 0020                 lxi     D,32
1777    19                      dad     D       ; load cur rec addr
1778    77                      mov     M,A     ; next rec is now cur I/O rec
                        ;set dma address
1779    11 162C                 lxi     D,dskbuf ; 128 chr buffer addr
177C    0E1A                    mvi     C,dma   ; prepare to set dma
177E    CD 0005                 call    BDOS
                        ;write the record
1781    ED5B 171E               lded    FCBaddr
1785    0E15                    mvi     C,wnr   ;write next record
1787    CD 0005                 call    BDOS ; write buffer to disk
                        ;check for successful write
178A    FE00                    cpi     0           ; errors?
178C    CA 17BB                 jz      resDMA      ; No.
178F    21 17A5                 lxi     H,badwrmsg ; Yes.
1792    CD 1385                 call    prtmsg
1795    3A 1538                 lda     ERRDSK
1798    C641                    adi     'A'
179A    4F                      mov     C,A
179B    CD 1421                 call    CONOUT
179E    0E3A                    mvi     C,':'
17A0    CD 1421                 call    CONOUT
17A3    1816                    jmpr    resDMA
17A5    0D0A43616E74 badwrmsg: .asciz [cr][lf]'Cant write data to '
                        ;reset dma
17BB    11 0080      resDMA:    lxi     D,80h
17BE    0E1A                    mvi     C,dma
17C0    CD 0005                 call    BDOS
                        ;close file CONFILE.TXT
17C3    ED5B 171E               lded    FCBaddr
17C7    0E10                    mvi     C,close
17C9    CD 0005                 call    BDOS        ; close the file
17CC    FEFF                    cpi     0FFh        ; OK close?
17CE    C0                      rnz                 ; Yes. Return.
17CF    21 17A5                 lxi     H,badwrmsg; No. Load err msg
17D2    CD 1385                 call    prtmsg
17D5    3A 1538                 lda     ERRDSK
```

```
17D8    C641            adi     'A'
17DA    4F              mov     C,A
17DB    CD 1421         call    CONOUT
17DE    ØE3A            mvi     C,':'
17EØ    CD 1421         call    CONOUT
17E3    C9              ret                     ; and return.


                        ;----------
                        ;  Suboutine: BUFreset
                        ;
                        ; Reset variables, buffers, counters, and disk selects.
                        ;
17E4                    BUFreset:
                        ;reselect test disk
17E4    3A 1483                 lda     DSKNAME
17E7    0602                    mvi     B,2
17E9    CD 1832                 call    chaMAPbyt ; select test disk
                        ;restore the original bytes-per-read value
17EC    ED4B 1485               lbcd    BUFLEN
17FØ    CD 147B                 call    SETBYT
                        ;------------------
                        ;reset 128 chr disk buffer and reset counter
17F3    21 162C                 lxi     H,dskbuf
17F6    22 1628                 shld    bufchr  ; reset buffer for next 128
17F9    3E00                    mvi     A,Ø
17FB    32 162A                 sta     bufcnt  ; reset counter
                        ;----------
                        ;Check the error buffer to see if weve passed B000h.
17FE    2A 1535                 lhld    errbufadr ; get current err buf addr
1801    3EB0                    mvi     A,ØB0h
1803    BC                      cmp     H         ; have we passed B000h?
1804    F2 1820                 jp      resume    ; No, errbuf is not full.
                        ;Error buffer is full.  Reset it. Leave 'truncate' msg.
1807    21 7000                 lxi     H,erbuffer
180A    22 1535                 shld    errbufadr
180D    3EFF                    mvi     A,0FFh
180F    32 1537                 sta     errflg    ; turn on err buf write flag
1812    21 154E                 lxi     H,truncmsg
1815    CD 1385                 call    prtmsg    ; 'buffer truncated'
1818    3E00                    mvi     A,Ø
181A    32 1537                 sta     errflg    ; turn off err buf write flag
181D    CD 170F                 call    putEOF    ; label end of err buffer
                        ;----------
                        ;Rebalance the stack and resume testing
1820    E1              resume: pop H
1821    D1                      pop D
1822    C1                      pop B
1823    F1                      pop PSW
1824    C9                      ret


                        ;----------
                        ;Subroutine:    stoMAPbyt
                        ; Regs  in:     A = disk number of MAPbyte to store
                        ; Regs out:     none
```

```
                        ;Destroyed:     any and/or all registers

                        ; Save the current assignment for restoration.
1825                    stoMAPbyt:
1825    3A 0004                 lda     04      ; get current drive
1828    32 153A                 sta     orgdriv ; and save for restoration.
182B    CD 1473                 call    CPMmap  ; get current assignment
                        ; BIOS address of the byte returns in HL
182E    32 171D                 sta     MAPbyt  ; save it for later restoration
1831    C9                      ret             ; and return.

                        ;----------
                        ;Subroutine:    chaMAPbyt
                        ; Regs  in:     A = unit number to assign to
                        ;               B = CP/M device number to assign from
                        ; Regs out:     none
                        ;Destroyed:     any and/or all registers

                        ; Change the assignment of the CP/M drive number
                        ; (received in B) to that of a floppy disk on the
                        ; unit number received in the accumulator.
1832                    chaMAPbyt:
1832    F5                      push    PSW     ; Save unit number
1833    48                      mov     C,B
1834    CD 1443                 call    SELDSK  ; select unit 0 f
1837    CD 1473                 call    CPMmap  ; get current MAPbyte byte adir
                        ; Current unit 0 assignment (network,etc) returns in A
                        ; BIOS address of the byte returns in HL
183A    C1                      pop     B       ; Restore unit number in B
183B    3A 171C                 lda     DENSbyt ; Get floppy density code
183E    B0                      ora     B       ; A = A <or> B = New MAPbyte
183F    77                      mov     M,A     ; Store it in the BIOS
1840    C9                      ret

                        ;----------
                        ;Subroutine:  resMAPbyt
                        ; Regs  in:     A = disk number of MAPbyte to restore.
                        ; Regs out:     none
                        ;Destroyed:     any and/or all registers

                        ; Restore the error disk assignment
1841                    resMAPbyt:
1841    3A 153A                 lda     orgdriv ; Get our original disk
1844    4F                      mov     C,A     ; and put it in C
1845    CD 1443                 call    SELDSK  ; and select buffer-write unit.
1848    CD 1473                 call    CPMmap  ; get BIOSmap addr
                        ;Current unit assignment returns in A
                        ;BIOS address of the byte returns in HL
184B    3A 171D                 lda     MAPbyt  ; original assignment
184E    77                      mov     M,A     ; restore previous assignment
184F    C9                      ret             ; and return.
```

```
                        ;***********************************************
                        ;*                                             *
                        ;*  TIME PRINT-OUT DATA,DEFS, AND SUBROUTINES  *
                        ;*                                             *
                        ;***********************************************
                        ;----------(From Utility Program 'TIME'.)
                        ; Get the date from 44h,45h, & 46h
                        ; and print out in months (J-F),days
                        ; (1-31), and year (00-99).     D.Stein 09/05/80
                        ;
0044                    month   =       44h
0045                    day     =       45h
0046                    year    =       46h
                        ;
1850                    prttime:
                        ;print the date message
1850    21 197A                 lxi     H,datmsg
1853    CD 1385                 call    prtmsg
                        ;out the month,day and year (XXX-DD-YY)
1856    21 0044                 lxi     H,month
1859    7E                      mov     A,M     ;1-12 in A
185A    21 1992                 lxi     H,Janmsg
185D    FE01                    cpi     1       ;January?
185F    CA 18BD                 jz      prmnth
1862    21 1997                 lxi     H,Febmsg
1865    FE02                    cpi     2       ;February?
1867    CA 18BD                 jz      prmnth
186A    21 199C                 lxi     H,Marmsg
186D    FE03                    cpi     3       ;March?
186F    CA 18BD                 jz      prmnth
1872    21 19A1                 lxi     H,Aprmsg
1875    FE04                    cpi     4       ;April?
1877    CA 18BD                 jz      prmnth
187A    21 19A6                 lxi     H,Maymsg
187D    FE05                    cpi     5       ;May?
187F    CA 18BD                 jz      prmnth
1882    21 19AB                 lxi     H,Junmsg
1885    FE06                    cpi     6       ;June?
1887    CA 18BD                 jz      prmnth
188A    21 19B0                 lxi     H,Julmsg
188D    FE07                    cpi     7       ;July?
188F    CA 18BD                 jz      prmnth
1892    21 19B5                 lxi     H,Augmsg
1895    FE08                    cpi     8       ;August?
1897    CA 18BD                 jz      prmnth
189A    21 19BA                 lxi     H,Sepmsg
189D    FE09                    cpi     9       ;September?
189F    CA 18BD                 jz      prmnth
18A2    21 19BF                 lxi     H,Octmsg
18A5    FE0A                    cpi     10      ;October?
18A7    CA 18BD                 jz      prmnth
18AA    21 19C4                 lxi     H,Novmsg
18AD    FE0B                    cpi     11      ;November?
18AF    CA 18BD                 jz      prmnth
18B2    21 19C9                 lxi     H,Decmsg
```

```
18B5    FE0C            cpi     12      ;December?
18B7    CA 18BD         jz      prmnth
                ;error trap
18BA    21 19CE         lxi     H,XXXmsg
18BD    CD 1385     prmnth: call    prtmsg
                ;
18C0    21 0045     prtday: lxi     H,day
18C3    7E              mov     A,M
18C4    CD 1950         call    cvtbcd
                        ;is leading 0 necessary?
18C7    FE0A            cpi     10
18C9    F2 18D3         jp      aa2
18CC    F5              push    PSW
18CD    0E30            mvi     C,'0'   ; out a leading 0
18CF    CD 1421         call    CONOUT
18D2    F1              pop     PSW
                        ;
18D3    CD 1959     aa2:    call    prtbyt  ; out the day
18D6    0E2D            mvi     C,'-'
18D8    CD 1421         call    CONOUT  ;
                ;
18DB    21 0046         lxi     H,year
18DE    7E              mov     A,M
18DF    CD 1950         call    cvtbcd
                        ;is leading 0 necessary?
18E2    FE0A            cpi     10
18E4    F2 18EE         jp      aa3
18E7    F5              push    PSW
18E8    0E30            mvi     C,'0'
18EA    CD 1421         call    CONOUT
18ED    F1              pop     PSW
                        ;
18EE    CD 1959     aa3:    call    prtbyt  ; date is all out.
                ;space over
18F1    21 1988         lxi     H,space5
18F4    CD 1385         call    prtmsg
                ;----------
                ; Print the real time (in hours and minutes and seconds)

                ; as represented by these bytes:
0041                    seconds =       41h
0042                    minutes =       42h
0043                    hours   =       43h
                ;----------
                ; Print the time on the DSC/3 or DSC/4.
                ; This program is used in conjunction with the TIMERopt
                ; option of the BIOS on the DSC/3 and DSC/4.
                ; The BIOS maintains the time in locations 40h-43h.
                ;
                ; Print the time message
18F7    21 1981         lxi     H,timmsg
18FA    CD 1385         call    prtmsg
                ; Print the hours
18FD    21 0043         lxi     H,hours
1900    7E              mov     A,M     ;get hours in A
```

```
1901    CD 1950             call    cvtbcd
                            ;is leading 0 necessary?
1904    FE0A                cpi     10
1906    F2 1910             jp      bb1
1909    F5                  push    PSW
190A    3E30                mvi     A,'0'
190C    CD 1421             call    CONOUT
190F    F1                  pop     PSW
                            ;
1910    CD 1959     bb1:    call    prtbyt
1913    0E3A                mvi     C,':'
1915    CD 1421             call    CONOUT
                    ;
                    ; Print the minutes
1918    21 0042     prtmin: lxi     H,minutes
191B    7E                  mov     A,M     ;get minutes in A
191C    CD 1950             call    cvtbcd
                            ;is leading 0 necessary?
191F    FE0A                cpi     10
1921    F2 192B             jp      bb2
1924    F5                  push    PSW
1925    0E30                mvi     C,'0'
1927    CD 1421             call    CONOUT
192A    F1                  pop     PSW
                            ;
192B    CD 1959     bb2:    call    prtbyt
192E    0E3A                mvi     C,':'
1930    CD 1421             call    CONOUT
                    ;
                    ; Print the seconds
1933    21 0041     prtsec: lxi     H,seconds
1936    7E                  mov     A,M
1937    CD 1950             call    cvtbcd
                            ;is leading 0 necessary?
193A    FE0A                cpi     10
193C    F2 1946             jp      bb3
193F    F5                  push    PSW
1940    0E30                mvi     C,'0'
1942    CD 1421             call    CONOUT
1945    F1                  pop     PSW
                            ;
1946    CD 1959     bb3:    call    prtbyt
1949    21 198F             lxi     H,crlf
194C    CD 1385             call    prtmsg  ; space down
194F    C9                  ret             ; finished printing time
                    ;----------
                    ; Convert binary to BCD
                    ;  Regs in:   A = byte to be converted
                    ;  Regs out:  A = byte, in BCD format
                    ;  Destroyed: B
1950                cvtbcd:
1950    B7                  ora     A
1951    C8                  rz
1952    47                  mov     B,A
1953    AF                  xra     A
```

```
1954    3C              ..1:    inr     A
1955    27                      daa
1956    10FC                    djnz    ..1
1958    C9                      ret
                        ;----------
                        ; Print a byte on the console
                        ;  Regs in:   A = byte to be printed
                        ;  Regs out:  none
                        ;  Destroyed: A
1959                    prtbyt:
1959    F5                      push    PSW
195A    0F                      rrc
195B    0F                      rrc
195C    0F                      rrc
195D    0F                      rrc
195E    E60F                    ani     0Fh     ; don't print leading zeros
1960    2803                    jrz     ..1
1962    CD 1966                 call    prtnbl
1965    F1              ..1:    pop     PSW
1966    E60F            prtnbl: ani     0Fh
1968    C630                    adi     '0'
196A    FE3A                    cpi     '9'+1
196C    4F                      mov     C,A     ; get the chr in C
196D    F5                      push    PSW
196E    DC 1421                 cc      CONOUT
1971    F1                      pop     PSW
1972    D8                      rc
1973    C607                    adi     'A'-('9'+1)
1975    4F                      mov     C,A
1976    CD 1421                 call    CONOUT
1979    C9                      ret
                        ;----------
                        ; Messages
197A    446174652020    datmsg: .asciz  'Date  '
1981    54696D652020    timmsg: .asciz  'Time  '
1988    202020202020    space5: .asciz  '      '
198F    0D0A00          crlf:   .asciz  [cr][lf]
1992    4A616E2D00      janmsg: .asciz  'Jan-'
1997    4665622D00      febmsg: .asciz  'Feb-'
199C    4D61722D00      marmsg: .asciz  'Mar-'
19A1    4170722D00      aprmsg: .asciz  'Apr-'
19A6    4D61792D00      maymsg: .asciz  'May-'
19AB    4A756E2D00      junmsg: .asciz  'Jun-'
19B0    4A756C2D00      julmsg: .asciz  'Jul-'
19B5    4175672D00      augmsg: .asciz  'Aug-'
19BA    5365702D00      sepmsg: .asciz  'Sep-'
19BF    4F63742D00      octmsg: .asciz  'Oct-'
19C4    4E6F762D00      novmsg: .asciz  'Nov-'
19C9    4465632D00      decmsg: .asciz  'Dec-'
19CE    5858582D00      XXXmsg: .asciz  'XXX-'

                        .end
```

+++++ SYMBOL TABLE +++++

AA2	18D3	AA3	18EE	APRMSG	19A1	AUGMSG	19B5
BAD1MS	0A8D	BAD2MS	0A9A	BADTMS	0A73	BADTRA	0A41
BADWRM	17A5	BB1	1910	BB2	192B	BB3	1946
BDOS	0005	BEGIN	0150	BIOSBU	153B	BIOSRT	004E
BUFCHR	1628	BUFCNT	162A	BUFLEN	1485	BUFREA	000A
BUFRES	17E4	CHAMAP	1832	CHR	014D	CHROUT	1433
CLOSE	0010	CLRERR	0209	COMLIN	0080	COMMSG	1011
COMPAR	0D8E	COMPER	09B3	COMPMS	09BC	CONBUF	12DC
CONIN	1407	CONOUT	1421	CONST	13FF	CONTMS	0FF3
CPMDSK	0360	CPMMAP	1473	CR	000D	CREATE	0016
CRLF	198F	CRLFMS	1392	CUR1MS	0BF2	CUR2MS	0C06
CVTBCD	1950	DATAER	081A	DATAMS	0820	DATMSG	197A
DAY	0045	DDBYT	0040	DECMSG	19C9	DENMSG	0390
DENSBY	171C	DENSER	092F	DENSMS	0935	DFCB	005C
DISKIT	1746	DMA	001A	DRIMSG	03F1	DSKASK	0163
DSKBUF	162C	DSKERR	0315	DSKFLG	162B	DSKFUN	14AF
DSKMSG	032B	DSKNAM	1433	DSKQST	05CB	DTYPEE	0A3F
ENDTER	085D	ENDTMS	0853	EOF	001A	ERBUFF	7000
ERRBUF	1535	ERRDSK	1538	ERREXI	02DD	ERRFLG	1537
ERRMSG	0434	ERRRTN	0763	ERRTAB	07A9	ERRTRY	14AB
EXIT	02D1	EXITMS	0305	FCB	1720	FCBADD	171E
FDMSG	0BDB	FEBMSG	1997	FILLPA	0DA7	FILLTA	02F5
FINDDE	0223	FINDED	013D	FINDTD	025A	FIXFIL	0DC9
FIXFLG	14AD	FIXMSG	0DDA	FIXTES	1142	FXABOR	12AD
FXDISK	1141	FXPRTB	12E2	FXRDMS	110F	FXREAD	1293
FXRETU	12B0	FXSTAR	113B	FXTRAC	1140	FXWRIT	1279
FXWRMS	1127	GETCPM	1353	GETDSK	1333	HDFILE	045C
HEADMS	13E5	HELPMS	155D	HOME	143B	HOURS	0043
IDERR	088D	IDMSG	0393	INCERR	0AAD	INITMS	059B
IOERR	0999	IOMSG	099F	JANMSG	1992	JULMSG	19B0
JUNMSG	19AB	KILL	0013	LF	000A	LINTAB	2000
LOGDSK	000E	LOGOUT	0B7C	LOOPMS	05E0	LOOPS	1437
MADRER	096B	MADRMS	0971	MAKHBY	12F8	MAPBYT	171D
MARMSG	199C	MAXERR	0104	MAYMSG	19A6	MINUTE	0042
MONTH	0044	NOVMSG	19C4	OCTMSG	19BF	OFFMSG	0FC0
OFLOWM	0AE5	OPEN	000F	ORGDRI	153A	ORUNER	08A1
ORUNMS	08A7	OUTCHR	16EA	PRMNTH	18BD	PROTER	08F9
PROTMS	0902	PRT16	13BF	PRT1UN	0B3F	PRTA	13A2
PRTBUF	16DC	PRTBYT	1959	PRTCUR	0C4C	PRTDAY	18C0
PRTDKI	0B0F	PRTMIN	1918	PRTMSG	1385	PRTNBL	1966
PRTNIB	13AF	PRTSEC	1933	PRTSUM	0C35	PRTTIM	1350
PUTBUF	1326	PUTEOF	170F	RANSEE	1488	RATIOC	0CE2
RD1UNI	06DA	RDBUFF	5000	RDERMS	079A	RDERR	0737
RDLNMS	05FA	RDPARA	14B3	RDRNMS	0622	RDTAB	2093
RDTEST	069F	RDTRK	0D5A	RDWRTR	0B0B	READ	1433
READCO	122D	RESDMA	17BB	RESET	000D	RESMAP	1841
RESUME	1820	RETDSK	1391	RETMSG	1391	RETRY	000A
REVISI	0003	SAVNBL	0144	SDBYT	0000	SECOND	0041
SECTER	08D2	SECTMS	08D8	SEEKER	07D0	SEEKMS	07D6
SELDSK	1443	SEPMSG	19BA	SERROR	149A	SETBYT	147B
SETDMA	145B	SETSEC	1453	SETTRK	144B	SETUP	012C
SPACE5	1988	STACK	7000	START	0105	STNDBY	0372
STOBYT	16AC	STODEN	024B	STOMAP	1325	STYPEE	0A06
SUMA	0C1A	SUMANM	0C20	SUMATR	0C22	SUMB	0C2C
SUMC	0C3B	SYNCER	07EE	SYNCMS	07F4	TERMLO	0BB8

+++++ SYMBOL TABLE +++++

```
TERROR 1489        TEST   04DD        TIMMSG 1981        TMPDSK 1539
TMPFLG 113F        TRACER 0830        TRACK  1484        TRACMS 0836
TRUNCM 154E        TSTLOO 04E4        TSTMSG 03BB        TSTSTR 050D
TSTUNI 14B5        TTYPEE 09CC        VERMSG 0CCE        VERSIO 0002
WAITCR 1371        WAITMS 1395        WBCHR1 063E        WBCHR2 063F
WBOOT  0001        WNR    0015        WR1UNI 0686        WRBUFF 3000
WRERMS 078A        WRERR  0709        WRITE  146B        WRLNMS 05E6
WRPARA 14B1        WRRNMS 060E        WRSTBY 0103        WRTAB  204C
WRTBUF 16CF        WRTEST 064B        WRTRK  0D20        WRWRST 0636
XXXMSG 19CE        YEAR   0046        .BLNK. 0000:03 X   .DATA. 0000*   X
.PROG. 0000'   X
```