RANDOM NUMBER SUBROUTINE FOR IBM 1620
FORTRAN II-D UNDER MONITOR I

DR. JOHN MANIOTES
COMPUTER TECHNOLOGY DEPT.
PURDUE UNIVERSITY
CALUMET CAMPUS
HAMMOND, IN 46323

RANDOM NUMBER SUBROUTINE FOR IBM 1620
FORTRAN II-D UNDER MONITOR I

Donald L. Fink

September 15, 1964

Direct Inquiries To:

Donald L. Fink
IBM Corporation
2330 Saint Paul Street
Baltimore, Maryland 21218

Modifications or revisions to this program, as they occur, will be announced in the appropriate Catalog of Programs for IBM Data Processing Systems. When such an announcement occurs, users should order a complete new program from the Program Information Department.

DECK KEY

Deck 1	SPS Source Deck - sequence # in cc 1-5, 72 cards including 9 control cards
Deck 2	Condensed 1620 Object Deck - sequence # in cc 76-80, 18 cards including 6 control cards
Deck 3	FORTRAN Test Program - sequence # in cc 79-80, 46 cards

TABLE OF CONTENTS

I. PROGRAM BRIEF

A. Purpose - A program to be incorporated into FORTRAN II-D under Monitor I disk system as a relocatable library subroutine. Pseudo-random numbers from either a rectangular or a standard normal distribution can be generated.

B. Method - The technique used is the power residue method. Each random number serves as the multiplicand for the next number. The formula is:

$$R.N._{n+1} = K \times R.N._{n} \ (\text{mod. } 10^{10})$$

where K is a 10 digit constant multiplier defined in the subroutine. Only the 10 low order digits of the products are retained as the random number. (See technique discussion in Statistical Forecasting for Inventory Control by Robert G. Brown, McGraw-Hill Book Co., Inc., New York, 1959, p. 164.)

C. Restrictions - This subroutine should only be used in programs in which the mantissa length of floating point numbers is 8 (i.e., F=8). There is no restriction on the size of fixed point numbers. If the F size restriction can not be met, a separate FORTRAN II program can be written to punch a deck of random numbers which can be read as input into the main program as they are needed.

D. Accuracy - The power residue method multiplies a starting value by a constant prime number to produce a product. It is only the low order half of this product which is used as the random number, i.e., P_{11} to P_{20} below.

$$P_1P_2P_3P_4P_5P_6P_7P_8P_9P_{10}\underbrace{P_{11}P_{12}P_{13}P_{14}P_{15}P_{16}P_{17}P_{18}P_{19}P_{20}}_{\text{random number}}$$

The rightmost few digits of the product cycle in an easily detectable pattern and, hence, can not very well be called random. The "best" random digit is at P_{11}, and it is this point from which the random number is taken. A cycle of 50 million 10 digit numbers is expected before the sequence repeats. A more thorough discussion of the manipulation of this number is presented in the next section.

E. Machine Configuration - Specifications for FORTRAN II-D must be met. The only special feature used in the subroutine is indirect addressing, which is a FORTRAN II requirement.

F. Program Requirements - This subroutine uses 574 positions of core. It contains its own normalizing routine to store a rectangular (.00000000 - .99999999) or normal (generally -3.5 to +3.5) number in proper floating point form in FAC (a pseudo accumulator in which FORTRAN deposits numerical results.)

G. Source Language - This subroutine was written in SPS and assembled using SPS II-D. The condensed object deck is provided along with the source deck.

H. Program Execution Time - Times in the subroutine to generate random numbers on a model I 1620 are as follows: rectangular distribution, 25 msec; normal distribution, 250 msec. During this time the argument is obtained, a number is generated and normalized, and the result is placed in FAC. Times will vary slightly depending upon the number of leading zeroes in the unnormalized result.

I. Check-Out Status - This routine was tested and is being used by one installation at the time of submission of this program. Results of tests agree exactly with results from 1620-7.0.021, a random number generator subroutine written for FORTRAN W/FORMAT.

J. Sample Problem Running Time - The execution time of the FORTRAN object program is approximately 6 minutes on the 1620 Model 1.

K. Comments - This program and its documentation were written by an IBM employee. It was developed for a specific purpose and submitted for general distribution to interested parties in the hope that it might prove helpful to other members of the data processing community. The program and its documentation are essentially in the author's original form. IBM serves only as the distribution agency in supplying this program. Questions concerning the use of the program should be directed to the author's attention.

II. DETAILED PROGRAM DESCRIPTION

A. Rectangular Numbers - In FORTRAN a relocatable library subroutine is called by mentioning its name and the argument upon which it is to operate, e.g., SQRT(Y) or LOGF(2.621). In 1620 FORTRAN II-D the address of the argument is placed in the 5 positions of memory immediately preceding the location of the first instruction of the subroutine. The subroutine itself is placed, along with other subroutines called by the mainline program, at the end of the object code which FORTRAN generates. (See pp. 133-134 of Monitor I Reference Manual, C26-5739-2, for rules on subroutine operations.)

The power residue method of generating pseudo-random numbers requires a "seed", or starting value, to begin producing numbers. In this routine the "seed" is a 3 digit number, say, 231, in which case the user might write X=RAND(.231). The argument is stored in floating pointform as $\bar{2}310000000$. This is converted by the subroutine into $\bar{2}310000001$, which serves as a 10 digit multiplicand to be multiplied by $\bar{1}977326743$, a 10 digit prime number defined in the subroutine. The 10 low order digits of the 20 digit product are retained. In this example they are 8307326743.

This number is both the random number and the multiplicand for the next desired number. It must now be placed in FAC in correct floating point form. The 2 low order digits are replaced by the exponent and the number is placed in FAC as $\bar{8}307326700$, which is .83073267. Should the low order half of the product read 0071622343, the subroutine would alter its form to $\bar{0}071622300$ and then begin to delete leading zeroes and adjust the exponent in the normalizing routine. Finally the normalized result $\bar{7}162230002$ is obtained in FAC. This is, of course, $.716223 \times 10^{-2}$ or .00716223.

B. Normal Numbers - Normally distributed numbers are generated from rectangular numbers in an interesting fashion. The Central Limit Theorem of statistics states that for a series of independent, identically distributed variates $x_1, x_2, \ldots x_n$ with mean μ and variance σ^2, the statistic

$$S = \frac{\bar{x} - \mu}{\sigma/\sqrt{n}} \qquad \left(\text{where } \bar{x} = \frac{\sum x_i}{n}\right)$$

or

$$S = \frac{\sum x - n\mu}{\sqrt{n}\,\sigma}$$

becomes normally distributed with mean 0 and variance 1 as $n \to \infty$. For practical purposes very good results can be obtained with a modest sample size.

For a rectangularly distributed variable with limits of 0 and 1, the mean and variance of this variable can be shown to be .5 and 1/12 respectively. If a sample of 12 such independent numbers is taken, this sample will have an expected value of 6 and a variance of 1 since the variance of a sum of independent variables is equal to the sum of their variances. Since a rectangularly distributed random number has limits of 0 and 1 and since these numbers are theoretically independent, the statistic

$$S = \frac{\sum_{1}^{12} R.N._i - 12(.5)}{\sqrt{12}\ (1/\sqrt{12})}$$

$$= \sum_{1}^{12} R.N._i - 6.0$$

can be expected to be approximately normally distributed with mean 0 and variance 1. Therefor, it is seen that normal numbers can be produced by summing 12 rectangular numbers and subtracting 6.0. It is possible to produce both positive and negative numbers ranging from -6 to +6 with this technique. The theoretical normal distribution has limits of $-\infty$ and $+\infty$, but, for all practical purposes, numbers outside the range of -3.5 to +3.5 are extremely rare. Tests of normality of a distribution of pseudo-random normal numbers have shown this generation technique to give quite satisfactory results.

III. SUBROUTINE CALLING SEQUENCE

The object deck for the subroutine is provided with name cards which allow the subroutine to be called RAND or RANDF in keeping with customary FORTRAN subroutine naming rules. Every time a statement mentioning RAND or RANDF is encountered, a new number will be generated. A positive argument signifies a rectangularly distributed number, e.g., RAND(.179); a negative argument, a normally distributed number, e.g., RANDF(-.179).

As long as the argument in the calling statement does not change, the subroutine uses the last random number generated as the multiplicand for the next. This is done by saving the 3 digit argument in the subroutine and comparing it with the argument found each time the subroutine is called. Should the argument change, a new sequence is begun as defined in the preceding section of this write-up. (NOTE: .179 and -.179 are <u>not</u> the same arguments.)

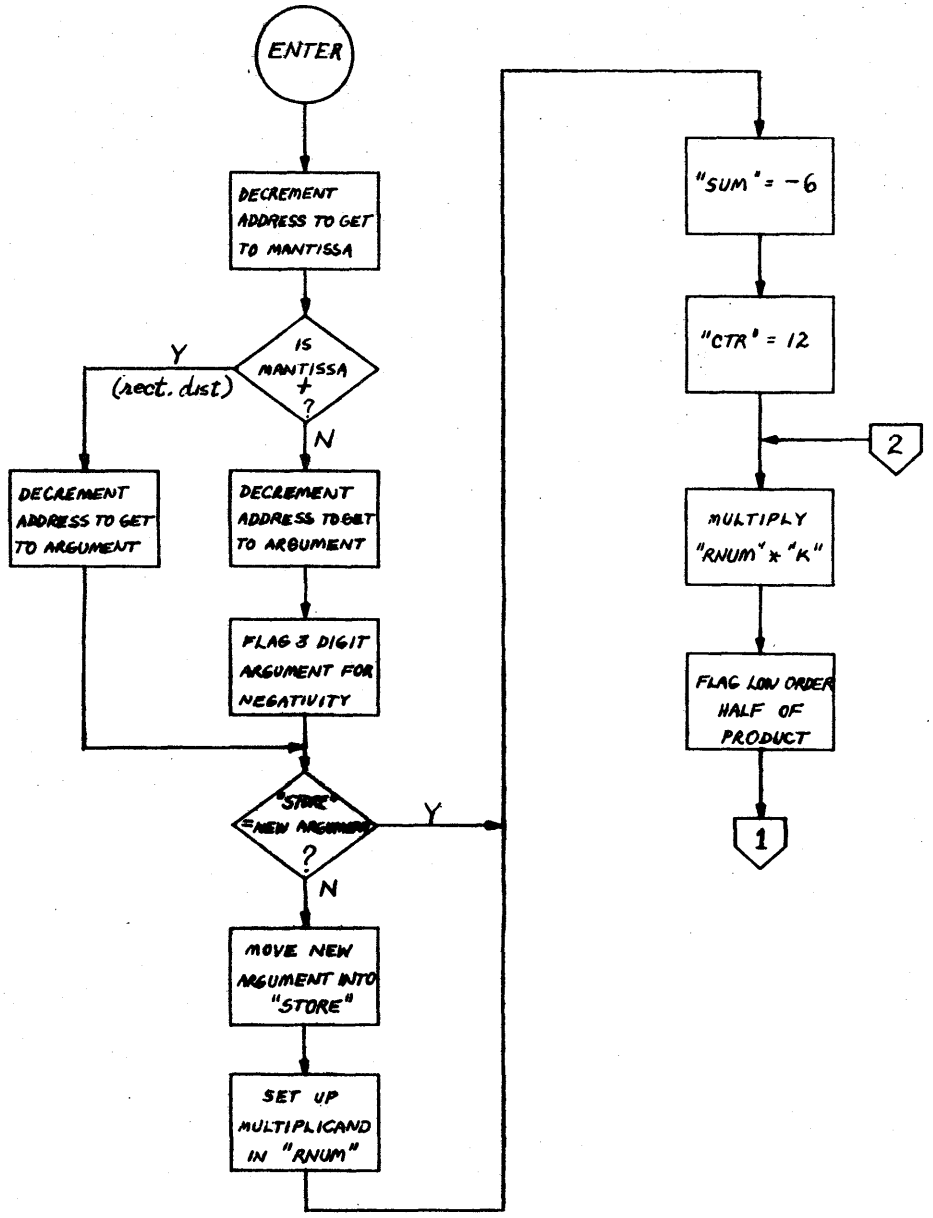
The example below will generate 50 rectangular numbers from one sequence, and then initiate a new sequence of 100 normal numbers.

```
      DO 10   I = 1, 50
10    A(I) = RANDF (.337)
      DO 20   I = 1, 100
20    B(I) = RAND (-.113)
```

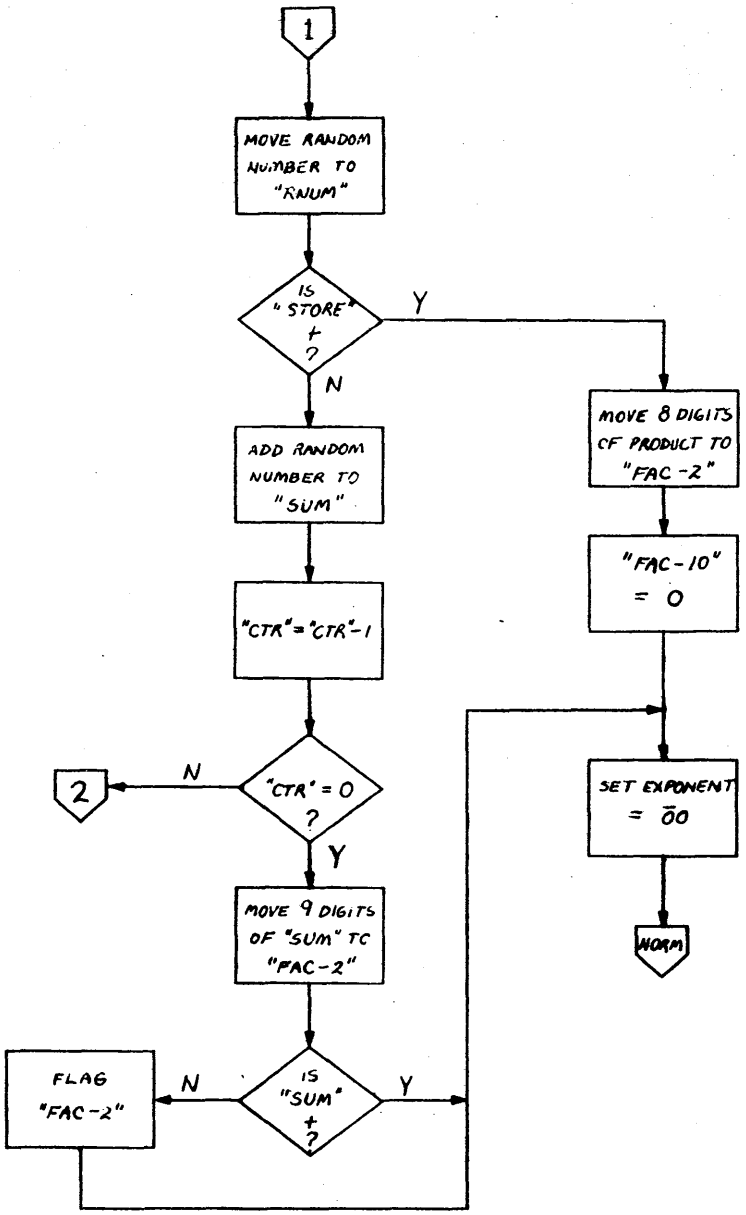
The following example is frequently seen in simulation problems. Suppose copper anodes are normally distributed with a mean of 250 lbs. and a standard deviation of 3 lbs. Simulated sampling of such a distribution can be obtained by the FORTRAN expression

```
WEIGHT = RAND (-.707) * 3.0  +  250.0
```

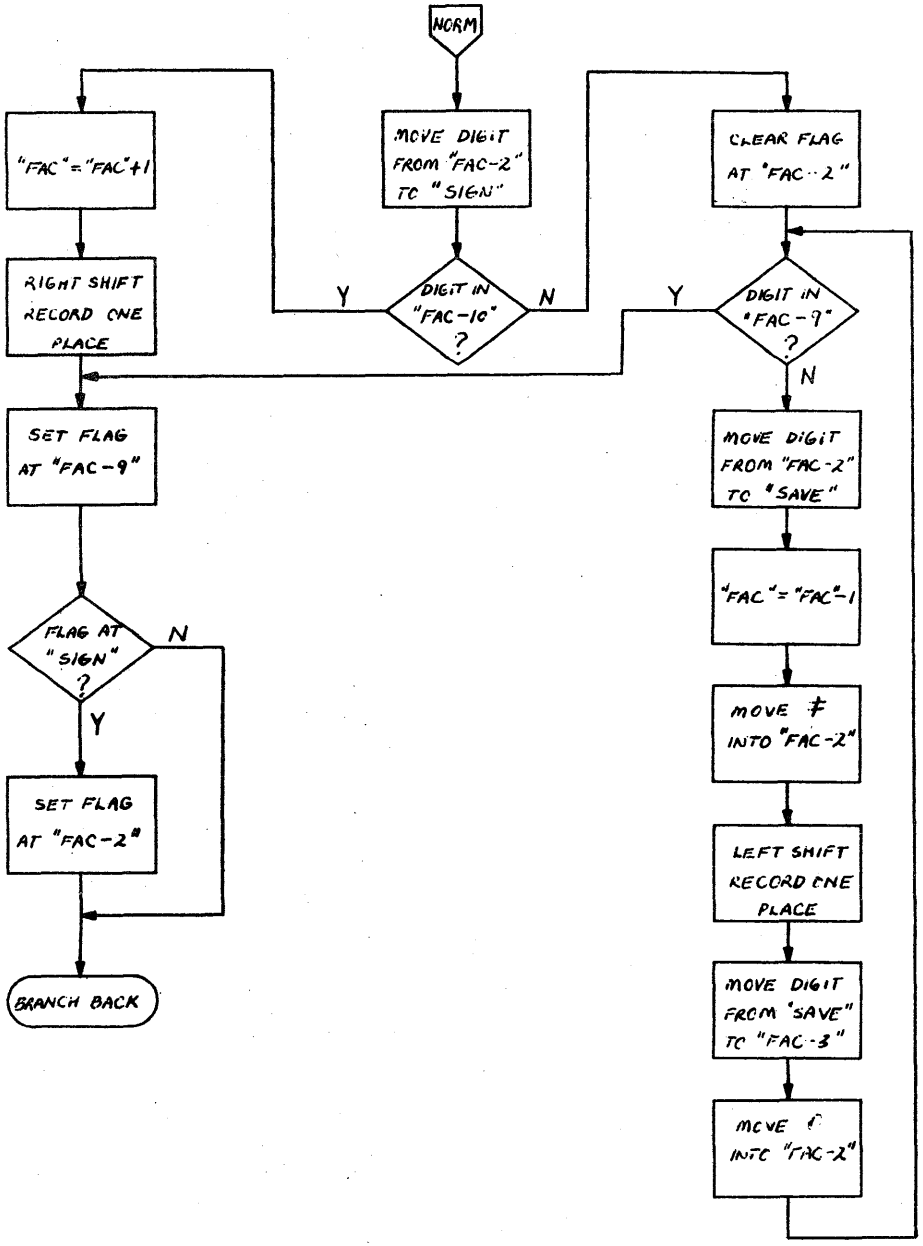
IV. RAND SUBROUTINE PROGRAM FLOWCHART



PROGRAM FLOWCHART (CONT.)



PROGRAM FLOWCHART (CONT.)



V. OPERATING INSTRUCTIONS

A. Console Switch Settings - Same as FORTRAN II-D requirements.

B. Loading Subroutine onto Disk - A condensed object deck with necessary control cards is provided for the user (deck 2). The structure of this deck is seen on p. 135 of the Monitor I Reference Manual. A monitor call and JOB card are also provided. The *DLOAD card (4th card) lists the subroutine name RAND in cc. 7-10. This can be changed to any name the user desires. The DIM number for this subroutine is 0026 in cc. 17-20 of the same card. If the user has added or deleted library subroutines previously, this number may have to be changed accordingly. (See rules on p. 135 of Monitor I Reference Manual.) The last card of the object deck contains the alternate name by which this subroutine may be called, RANDF, subject to the user's preference. The DIM number is also 26 in cc. 14-15. This may have to be changed as mentioned above. A complete listing of the object deck appears in another section of this write-up. To load the object deck merely depress the LOAD key at the card reader.

VI. SAMPLE PROBLEM

This sample problem generates and prints 10 rectangularly distributed random numbers and 10 normally distributed random numbers. It then generates 500 rectangular numbers and tallies a distribution of these numbers with .0 - .09999999 as case 1, .1 - .19999999 as case 2, ... , .9 - .99999999 as case 10. Finally 500 normal numbers are generated and tallied with those below -3.0 as case 1, -3.0 to -2.5000001 as case 2, ... , 2.5 to 2.9999999 as case 13, 3.0 or greater as case 14. At the conclusion of the sample problem "STOP" is printed at the console typewriter. The results of the sample problem appear in the Listings section which follows.

The user should compile and execute the FORTRAN test problem on his own 1620-1311 system. This program tests the correct placement of the RAND subroutine onto the 1311 disk. (NOTE: the random number generator subroutine must be stored on disk before the program will execute successfully.)

VII. PROGRAM LISTINGS

The following program listings are presented:

1) SPS source for RAND subroutine with control cards

2) SPS condensed object deck with control cards

3) FORTRAN program using RAND subroutine as sample problem (no data required)

4) Results of sample problem

RAND SUBROUTINE - SPS SOURCE

```
340003200701360003200702490240251196361130010 2
ZZJOB 5
ZZSPS
*LIBR
*NAME RAND
*ASSEMBLE RELOCATABLE
*STORE RELOADABLE
*ID NUMBER 0026
* OUTPUT CARD
 1040       DSA RAND
 1070RAND   SM  ARGADD,2,,DECREMENT ADDRESS TO GET AT MANTISSA OF ARGUMENT
 1080       BNF COMP,ARGADD,11,BRANCH TO COMP IF MANTISSA IS POSITIVE
 1090       SM  ARGADD,5,,DECREMENT ADDRESS TO GET AT 3 DIGIT ARGUMENT
 1110       SF  ARGADD,,6,FLAG LOW ORDER DIGIT OF ARGUMENT SINCE NEGATIVE
 1120       B   COMP&12
 1130       DORG*-3
 1140COMP   SM  ARGADD,5,,DECREMENT ADDRESS TO GET AT 3 DIGIT ARGUMENT
 1150       C   STORE,ARGADD,11,COMPARE OLD ARGUMENT WITH NEW
 1160       BE  SUMSET,,,IF EQUAL CONTINUE CHAIN OF R.N./S
 1180       TF  STORE,ARGADD,11,SINCE UNEQUAL PUT NEW ARGUMENT INTO STORE
 1190       TF  RNUM,CONST,,LOAD ZEROES AND 1 INTO RNUM TO FORM MULTIPLICAND
 1200       TF  RNUM-7,STORE,,LOAD 3 DIGIT ARGUMENT UNTO RNUM
 2030       CF  RNUM-7,,,REMOVE FLAG IF IT EXISTS
 2040SUMSETTF  SUM,NEGSIX,,SET SUMMING AREA TO -6
 2050       TFM CTR,12,10,SET COUNTER TO 12
 2070CALC   M   RNUM,K,,DEVELOP PRODUCT
 2080       SF  00090,,,FLAG HIGH ORDER END OF LOW ORDER HALF OF PRODUCT
 2090       TF  RNUM,00099,,MOVE RANDOM NUMBER TO RNUM
 2110       BNF SINGLE,STORE,,BRANCH IF RECTANGULAR DIST. NUMBER DESIRED
 2120       A   SUM,00099,,ADD NUMBER TO SUM
 2130       SM  CTR,1,10,DECREMENT COUNTER BY 1
 2140       BNZ CALC,,,RETURN IF 12 NUMBERS NOT YET SUMMED
 2150       TF  FAC-2,SUM-2,,MOVE OUT 9 HIGH ORDER DIGITS
 2160       BNF SINGLE&24,SUM,,BRANCH IF RANDOM NORMAL NUMBER IS POSITIVE
 2170       SF  FAC-2,,,OTHERWISE SET FLAG FOR NEGATIVITY
 2180       B   SINGLE&24
 2190       DORG*-3
 3030SINGLETF  FAC-2,00097,,LOAD RECT. DIST. NUMBER INTO FAC
 3040       TDM FAC-10,0,11
 3050       TFM FAC,0,10,SET EXPONENT EQUAL TO 00
 3080NORM   TD  SIGN,FAC-2,,THIS IS NORMALIZING ROUTINE USED BY BOTH
 3090       BD  ADDEXP,FAC-10
 3100       CF  FAC-2
 3110DIGTCKBD  FLGSET,FAC-9
 3120       TD  SAVE,FAC-2
 3130       SM  FAC,1,10
 3140       TD  FAC-2,RKMK
 3150       TR  FAC-9,FAC-8
 3160       TD  FAC-3,SAVE
 3170       TDM FAC-2,0
 3180       B   DIGTCK
```

RAND SUBROUTINE (CONT.)

```
3190        DORG*-3
4030ADDEXPAM   FAC,1,10
4040        TF   FAC-2,FAC-3
4050FLGSETSF   FAC-9
4060        BNF  *&24,SIGN
4070        SF   FAC-2
4080        BB
4090        DORG*-9
4110STORE DC   3,0,DIGTCK-3
4120RNUM  DS   10
4130CONST DC   8,1
4140K     DC   10,1977326743
4150NEGSIXDC   11,-60000000000
4160SUM   DS   11
4170CTR   DS   2,DIGTCK-1
4180SIGN  DS   1,FLGSET&7
4190SAVE  DS   1,FLGSET&8
4200RKMK  DC   1,@,FLGSET&9
4210FAC   DS   ,2492
1060ARGADDDS   ,RAND-1,FIELD TO CONTAIN ADDRESS OF ARGUMENT OF SUBR
4220        DEND1
```


RAND SUBROUTINE — OBJECT

```
34000321070136000320070249024025119636113000102
ZZJOB 5
ZZDUP 5
*DLOADRAND        0026                        005740000 1CS
00003050000 6Z000061M8J200005000002MM00062000 00NJ200005000005L20000N[illegible]0001
00054 1J2M900007400000Z000621L6J200005000005KM00367000 0NM60014601200R00000000000002
00098100K0003670000NK000533 00541K00052600367L30052600000K00057300552R0000000 00003
00158100J600369000J2KL00533005513200090000 00K6005330009 9MM002980036 7R00000000 0004
0021810 0K100573 0009 9J200369000 01M7001700120020024900057 1MM0032200573R00000000005
00278 1K43202490000 00M9003220000 0Z00298 1K42602490000971502482000 00R0000000000006
00322100160249200000K50049302490M30046202482330249000000M30048602483R00000000007
00382100K500494024901202492000012N0249000495310248302484 2N0248900494R0000000 0008
00442 1K415024900 0000M900370000 00Z004621K41102492000 01260249002489R0000000000009
00486 1M83320248300000MM00522004933202490000004200000000000Z003652030 00R00000000 010
005342K9[illegible]1J9773267430000000000 0Z004952012Z005746 00Z0000000000000000000-0011
R9999Z00000000000                                                           -0012
ZZDUP
*DFLIBRANDF   26
```

FORTRAN II-D SAMPLE PROBLEM

```
    SAMPLE PROBLEM - TEST OF RAND
    DIMENSION KASE%14¤
    PRINT 100
100 FORMAT %45HSOME RECTANGULARLY DISTRIBUTED RANDOM NUMBERS/¤
    DO 1 MAX # 1,10
    X # RAND%.431¤
  1 PRINT 101, X
101 FORMAT %F11.8¤
    PRINT 102
102 FORMAT %//40HSOME NORMALLY DISTRIBUTED RANDOM NUMBERS/¤
    DO 2 MAX # 1,10
    X # RAND%-.483¤
  2 PRINT 101, X
    PRINT 103
103 FORMAT %///39HDISTRIBUTION OF 500 RECTANGULAR NUMBERS//¤
    PRINT 104
104 FORMAT %13HCASE     TALLY/¤
    DO 19 I # 1,10
 19 KASE%I¤ # 0
    DO 20 MAX # 1,500
    X # RAND%.733¤
    I # X*10.0 & 1.0
 20 KASE%I¤ # KASE%I¤ & 1
    DO 21 I # 1,10
 21 PRINT 105, I, KASE%I¤
105 FORMAT %I3,I9¤
    DO 29 I # 1,14
 29 KASE%I¤ # 0
    DO 30 MAX # 1,500
    X # RAND%-.653¤
    IF %X&3.0¤ 31,32,32
 31 KASE%1¤ # KASE%1¤ & 1
    GO TO 30
 32 IF %X-3.0¤ 34,35,35
 35 KASE%14¤ # KASE%14¤ & 1
    GO TO 30
 34 I # 2.0*X & 8.0
    KASE%I¤ # KASE%I¤ & 1
 30 CONTINUE
    PRINT 106
106 FORMAT %///34HDISTRIBUTION OF 500 NORMAL NUMBERS//¤
    PRINT 104
    DO 36 I # 1,14
 36 PRINT 105, I, KASE%I¤
    STOP
    END
```


SAMPLE PROBLEM RESULTS

```
MAIN 07500 02568  LOADED
17 10068 00574  LOADED

SOME RECTANGULARLY DISTRIBUTED RANDOM NUMBERS

  .43073267
  .97729380
  .83078944
  .75868264
  .71957419
  .96206816
  .04570302
  .66006120
  .16054371
  .70594152

SOME NORMALLY DISTRIBUTED RANDOM NUMBERS

 1.53982260
  .64647868
 -.41166532
 -.79460932
  .33764668
 -.17489732
-1.49224130
 1.22561460
  .81867068
-1.87307330

DISTRIBUTION OF 500 RECTANGULAR NUMBERS

CASE     TALLY
  1        56
  2        49
  3        42
  4        50
  5        47
  6        49
  7        44
  8        58
  9        54
 10        51

DISTRIBUTION OF 500 NORMAL NUMBERS

CASE     TALLY
  1         0
  2         4
  3         6
  4        23
  5        47
  6        72
  7        95
  8       106
  9        64
 10        44
 11        28
 12         9
 13         2
 14         0
STOP
```

ASSEMBLY LISTING

```
01040          DSA   RAND                                          00004 00005 00006

01070 RAND     SM    ARGADD,2,,DECREMENT ADDRESS TO GET AT MANTISSA OF ARGUMENT
                                                                   00006 12 00005 00002
01080          BNF   COMP,ARGADD,11,BRANCH TO COMP IF MANTISSA IS POSITIVE
                                                                   00018 44 00062 00005
01090          SM    ARGADD,5,,DECREMENT ADDRESS TO GET AT 3 DIGIT ARGUMENT
                                                                   00030 12 00005 00005
01110          SF    ARGADD,,6,FLAG LOW ORDER DIGIT OF ARGUMENT SINCE NEGATIVE
                                                                   00042 32 00005 00000
01120          B     COMP+12                                       00054 49 00074 00000
01130          DORG  *-3                                           00062
01140 COMP     SM    ARGADD,5,,DECREMENT ADDRESS TO GET AT 3 DIGIT ARGUMENT
                                                                   00062 12 00005 00005
01150          C     STORE,ARGADD,11,COMPARE OLD ARGUMENT WITH NEW
                                                                   00074 24 00367 00005
01160          BE    SUNSET,,,IF EQUAL CONTINUE CHAIN OF R.N./S
                                                                   00086 46 00146 01200
01180          TF    STORE,ARGADD,11,SINCE UNEQUAL PUT NEW ARGUMENT INTO STORE
                                                                   00098 26 00367 00005
01190          TF    RNUM,CONST,,LOAD ZEROES AND 1 INTO RNUM TO FORM MULTIPLICAND
                                                                   00110 26 00533 00541
01200          TF    RNUM-7,STORE,,LOAD 3 DIGIT ARGUMENT UNTO RNUM
                                                                   00122 26 00526 00367
02030          CF    RNUM-7,,,REMOVE FLAG IF IT EXISTS             00134 33 00526 00000
02040 SUNSET   TF    SUM,NEGSIX,,SET SUMMING AREA TO -6            00146 26 00573 00562
02050          TFM   CTR,12,10,SET COUNTER TO 12                   00158 16 00369 00012
02070 CALC     M     RNUM,K,,DEVELOP PRODUCT                       00170 23 00533 00551
02080          SF    00090,,,FLAG HIGH ORDER END OF LOW ORDER HALF OF PRODUCT
                                                                   00182 32 00090 00000
02090          TF    RNUM,00099,,MOVE RANDOM NUMBER TO RNUM
                                                                   00194 26 00533 00099
02110          BNF   SINGLE,STORE,,BRANCH IF RECTANGULAR DIST. NUMBER DESIRED
                                                                   00206 44 00208 00367
02120          A     SUM,00099,,ADD NUMBER TO SUM                  00218 21 00573 00099
02130          SM    CTR,1,10,DECREMENT COUNTER BY 1               00230 12 00369 00001
02140          BNZ   CALC,,,RETURN IF 12 NUMBERS NOT YET SUMMED
                                                                   00242 47 00170 01200
02150          TF    FAC-2,SUM-2,,MOVE OUT 9 HIGH ORDER DIGITS
                                                                   00254 26 02490 00571
02160          BNF   SINGLE+24,SUM,,BRANCH IF RANDOM NORMAL NUMBER IS POSITIVE
                                                                   00266 44 00322 00573
02170          SF    FAC-2,,,OTHERWISE SET FLAG FOR NEGATIVITY
                                                                   00278 32 02490 00000
02180          B     SINGLE+24                                     00290 49 00322 00000
02190          DORG  *-3                                           00298
03030 SINGLE   TF    FAC-2,00097,,LOAD RECT. DIST. NUMBER INTO FAC
                                                                   00298 26 02490 00097
03040          TFM   FAC-10,0,11                                   00310 15 02482 00000
03050          TFM   FAC,0,10,SET EXPONENT EQUAL TO 00             00322 16 02492 00000
03060 NORM     TD    SIGN,FAC-2,,THIS IS NORMALIZING ROUTINE USED BY BOTH
                                                                   00334 25 00493 02490
03090          BD    ADDEXP,FAC-10                                 00346 43 00462 02482
03100          CF    FAC-2                                         00358 33 02490 00000
```



```
03110 DIGTCK   BD    FLGSFT,FAC-9                                  00370 43 00483 02483
03120          TD    SAVE,FAC-2                                    00382 25 00493 02490
03130          SM    FAC,1,10                                      00394 12 02492 00001
03140          TD    FAC-2,RNUK                                    00406 25 02490 00495
03150          TR    FAC-9,FAC-8                                   00418 31 02483 02484
03160          TD    FAC-3,SAVE                                    00430 25 02490 00494
03170          TRT   FAC-2,0                                       00442 15 02490 00000
03180          B     DIGTCK                                        00454 49 00370 00000
03190          DORG  *-3                                           00462
04030 ADDEXP   AM    FAC,1,10                                      00462 11 02492 00001
04040          TF    FAC-2,FAC-3                                   00474 26 02490 02489
04050 FLGSFT   SF    FAC-9                                         00486 32 02483 00000
04060          TNF   *+24,SIGN                                     00498 44 00522 00493
04070          SF    FAC-2                                         00510 32 02490 00000
04080          DB                                                  00522 42 00000 00000
04090          DORG  *-9                                           00524
04110 STORE    DC    3,0,DIGTCK-3                                  00367 00003 000
04120 RNUM     DS    10                                            00533 00010
04130 CONST    DC    8,1                                           00541 00008 00000001
04140 K        DC    10,1977326743                                 00551 00010 1977326743
04150 NEGSIX   DC    11,-60000000000                               00562 00011 0000000000
04160 SUM      DS    11                                            00573 00011
04170 CTR      DS    2,DIGTCK-1                                    00369 00002
04180 SIGN     DS    1,FLGSFT+7                                    00493 00001
04190 SAVE     DS    1,FLGSFT+8                                    00494 00001
04200 RNUK     DC    1,0,FLGSET+9                                  00495 00001 1
04210 FAC      DS    ,2492                                         02492 00000
01060 ARGADD   DS    ,RAND-1,FIELD TO CONTAIN ADDRESS OF ARGUMENT OF SUBR
                                                                   00005 00000
                                                                   00001
04220          BEND  1
```

END OF ASSEMBLY.
00574 CORE POSITIONS REQUIRED PLUS RELOCATION INCREMENT
00063 STATEMENTS PROCESSED

COMPUTER
TECHNOLOGY

Correction
7.0.057

1620
July 28, 1967

Attached is a Comments Sheet
which should be included in
the program writeup.

A user has recently pointed out that the following sequence of FORTRAN Statements will cause a check stop:

```
Y = RAND ( X )
X = X + 1.0
```

The second statement will fail to execute properly because the generator subroutine sets a flag withing the mantissa of the argument. This occurs only when X is negative, i.e., random normal numbers are being generated.

It is suggested that the use of arithmetic operations be avoided on the argument of the RAND subroutine. If such arithmetic is necessary (although the 'seed' is generally initialized and not altered during most runs) the following code may be used:

```
X = Z
Y = RAND ( X )
Z = Z + 1.0
X = Z
```