

AD-764 548

NAVAL POSTGRADUATE SCHOOL RANDOM  
NUMBER GENERATOR PACKAGE LLRANDOM

Gerard P. Learmonth, et al

Naval Postgraduate School  
Monterey, California

June 1973

DISTRIBUTED BY:

**NTIS**

National Technical Information Service  
U. S. DEPARTMENT OF COMMERCE  
5285 Port Royal Road, Springfield Va. 22151

NPS55LW73061A

# NAVAL POSTGRADUATE SCHOOL

## Monterey, California

AD 764548



NAVAL POSTGRADUATE SCHOOL  
RANDOM NUMBER GENERATOR PACKAGE LLRANDOM

by

G. P. Learmonth

and

P. A. W. Lewis

June 1973

Approved for public release; distribution unlimited.

Reproduced by  
NATIONAL TECHNICAL  
INFORMATION SERVICE  
U S Department of Commerce  
Springfield VA 22151

## DOCUMENT CONTROL DATA - R &amp; D

Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |  |                                                                             |                          |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|-----------------------------------------------------------------------------|--------------------------|
| 1. ORIGINATING ACTIVITY (Corporate author)<br><br>Naval Postgraduate School<br>Monterey, California 93940                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |  | 2a. REPORT SECURITY CLASSIFICATION<br><br>Unclassified                      |                          |
|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |  | 2b. GROUP                                                                   |                          |
| 3. REPORT TITLE<br><br>Naval Postgraduate School Random Number Generator Package LLRANDOM                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |  |                                                                             |                          |
| 4. DESCRIPTIVE NOTES (Type of report and, inclusive dates)<br><br>Technical Report                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |  |                                                                             |                          |
| 5. AUTHOR(S) (First name, middle initial, last name)<br><br>Peter A. W. Lewis and Gerard P. Learmonth                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |  |                                                                             |                          |
| 6. REPORT DATE<br><br>June 1973                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |  | 7a. TOTAL NO. OF PAGES<br><br>5748                                          | 7b. NO. OF REFS<br><br>6 |
| 8a. CONTRACT OR GRANT NO.<br><br>NRO42-284                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |  | 8a. ORIGINATOR'S REPORT NUMBER(S)                                           |                          |
| b. PROJECT NO                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |  |                                                                             |                          |
| c.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |  | 9a. OTHER REPORT NO(S) (Any other numbers that may be assigned this report) |                          |
| d.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |  |                                                                             |                          |
| 10. DISTRIBUTION STATEMENT<br><br>Approved for public release; distribution unlimited.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |  |                                                                             |                          |
| 11. SUPPLEMENTARY NOTES                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |  | 12. SPONSORING MILITARY ACTIVITY                                            |                          |
|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |  |                                                                             |                          |
| 13. ABSTRACT<br><br>This report is intended to describe an interim version of a computer program package for random number generation on the IBM System/360. The package, when called by a FORTRAN IV program, will deliver either a single value or an array (of specified size) of single precision uniformly, normally, or exponentially distributed pseudo-random deviates, or a single value or an array of uniformly distributed integers between 1 and $2^{31}-1$ . The package also has the ability (optional) to "shuffle" the pseudo-random numbers to obtain "better" statistical properties. |  |                                                                             |                          |

| 14<br>KEY WORDS          | LINK A |    | LINK B |    | LINK C |    |
|--------------------------|--------|----|--------|----|--------|----|
|                          | ROLE   | WT | ROLE   | WT | ROLE   | WT |
| Random number generator  |        |    |        |    |        |    |
| Pseudo-random numbers    |        |    |        |    |        |    |
| Normal distribution      |        |    |        |    |        |    |
| Exponential distribution |        |    |        |    |        |    |
| Shuffled random numbers  |        |    |        |    |        |    |
| Division simulation      |        |    |        |    |        |    |
| Congruential generator   |        |    |        |    |        |    |

NAVAL POSTGRADUATE SCHOOL  
Monterey, California

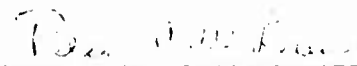
Rear Admiral M. B. Freeman  
Superintendent

M. U. Clauser  
Provost

ABSTRACT

This report is intended to describe an interim version of a computer program package for random number generation on the IBM System/360. The package, when called by a FORTRAN IV program, will deliver either a single value or an array (of specified size) of single precision uniformly, normally, or exponentially distributed pseudo-random deviates, or a single value or an array of uniformly distributed integers between 1 and  $2^{31}-1$ . The package also has the ability (optional) to "shuffle" the pseudo-random numbers to obtain "better" statistical properties.

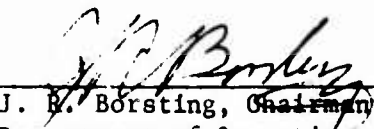
Prepared by:

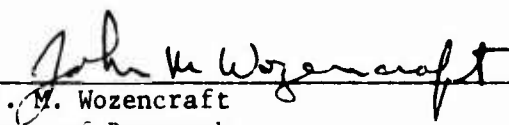
  
Peter A. W. Lewis\*  
Department of Operations Research  
and Administrative Sciences

  
Gerard P. Learmonth  
Computer Center

Approved by:

Released by:

  
J. B. Borsting, Chairman  
Department of Operations Research  
and Administrative Sciences

  
J. M. Wozencraft  
Dean of Research

\* Partial support by the Office of Naval Research through Contract NR042-284 is gratefully acknowledged.

NAVAL POSTGRADUATE SCHOOL  
RANDOM NUMBER GENERATOR PACKAGE LLRANDOM

|                                         |    |
|-----------------------------------------|----|
| I. Introduction                         | 1  |
| Some definitions                        | 1  |
| II. The Generator                       | 3  |
| A. Division simulation                  | 6  |
| B. Shuffling                            | 7  |
| C. Uniform (0.0,1.0) random numbers     | 9  |
| D. Normally distributed random deviates | 9  |
| E. Exponential distribution             | 12 |
| III. HOW TO USE THE PACKAGE             | 15 |
| A. Implementation                       | 18 |
| B. Future Enhancements                  | 20 |
| References                              | 21 |

Appendix of Program Listings

## I. Introduction.

Numerous random number generators have been proposed for the System/360. Several of these generators have been incorporated into the subroutine library here at the Computer Center. The adequacy of some of these generators has rested on the results of some rather weak tests for randomness; recent results in the literature have shown many of these generators to be very poor performers. This report will describe an interim version of a package for random number generation which has stood up under intensive statistical testing and is deemed to be very satisfactory for the System/360. (The statistical testing will be reported elsewhere.)

The package, when called by a FORTRAN IV program, will deliver either a single value or an array (of specified size) of single precision, uniformly, normally, or exponentially distributed pseudo-random deviates or a single value or an array of pseudo-random integers uniformly distributed between 1 and  $2^{31}-1$ . The package also has the ability (optional) to "shuffle" the pseudo-random numbers to obtain "better" statistical properties.

Further refinements will be made to this generator; however, it is now available for use in an interim version under the name LLRANDOM. Future versions will be announced through the W. R. Church Computer Center Newsletter. The changes envisioned will be internal and aimed at increasing speed and efficiency of coding. The actual numbers produced in future versions will remain the same as described here, as will the FORTRAN calling sequence.

Some definitions. By "random number generator" or "pseudo-random number generator" is meant an algorithm by which sequences of numbers

are produced which follow a given probability distribution and possess the appearance of randomness. Without attempting to address the still unresolved philosophical question of what a random sequence is, the underlined words above are the keys to random number generation on a digital computer. The term sequence implies that many random numbers must be produced serially from the algorithm. The user may need only a very few of these numbers, however we generally require that the algorithm be able to produce very many numbers. Distribution implies that we can associate a probability statement with the occurrence of each random number. The distribution is usually taken to be uniform, that is, within a given range the probability of occurrence of a given number is the same as for any other number in a similar range. If the algorithm produces, say,  $m$  distinct numbers then the probability of occurrence for any one of them is  $1/m$ .

Lastly, we speak of the appearance of randomness. As will be shown next, the actual implementation of the algorithm is a recurrence relation where each succeeding number is a function of the preceding number. True randomness would require independence of successive numbers; however, the algorithm generates a deterministic sequence. Algorithms for random number generation do, however, yield sequences which appear to be random, hence the term "pseudo-random numbers." It is this characteristic which is the subject of statistical testing, that is, one asks, "how random does the given sequence appear?"

The uniform random number generator which forms the basis of the package described here is a Lehmer congruential generator. The recurrence relation is given by

$$X_n \equiv A \cdot X_{n-1} + C \pmod{m}. \quad (1)$$

This generator produces integer random numbers between 1 and  $m$ . These integer values may then be transformed into real-valued numbers between 0.0 and 1.0 or into any desired distribution by an appropriate transformation.

## II. The Generator.

The recurrence relation given in equation (1) is actually called a "Lehmer mixed congruential generator." The term mixed comes from the fact that it involves a multiplication by a constant,  $A$ , plus an addition of a constant,  $C$ . The actual implementation used in LLRANDOM is called a "multiplicative," or "pure," congruential generator in that we take  $C = 0$ , giving

$$X_n \equiv A \cdot X_{n-1} \pmod{m}. \quad (2)$$

The field of positive integers is, of course, infinite. It is a reality of digital computers that only a finite number of positive integers are expressible. Specifically, we are limited to the word size of the System/360. This word size is 32 bits with one bit reserved for the algebraic sign; hence, an obvious choice for  $m$  is  $2^{31}$ . The product  $A \cdot X_{n-1}$  is formed by the System/360 in two adjacent registers yielding a result which may be as large as  $2^{63}$ . We must, however, reduce this product to a number less than or equal to  $2^{31}$ . The mod, or modulo, operation accomplishes this. The product  $A \cdot X_{n-1}$  is divided by  $2^{31}$  leaving a quotient which is some multiple of  $2^{31}$  and a remainder which is strictly less than  $2^{31}$ . It is this remainder which is the next pseudo-random number  $X_n$  in the equation (1).

On first examination it would appear that a full  $2^{31}$  numbers could be generated by the sequence (1). This is not the case, unless  $A$  and  $m$  are chosen properly. We define a term called the period which is the number of unique random numbers computable for a given choice of  $A$  and  $m$ . To illustrate the concept, assume we have a six-bit word with one bit for a sign. We then have  $m = 2^5 = 32$ . Choose  $A = 9$  and work through a sequence starting with  $X_0 = 1$ .

| Step $n$ | $X_{n-1}$ | $A \cdot X_{n-1}$ | $A \cdot X_{n-1} \pmod{2^5}$ |
|----------|-----------|-------------------|------------------------------|
| 1        | 1         | 9                 | 9                            |
| 2        | 9         | 81                | 17                           |
| 3        | 17        | 153               | 25                           |
| 4        | 25        | 225               | 1                            |
| 5        | 1         | 9                 | 9                            |
| .        | .         | .                 | .                            |
| .        | .         | .                 | .                            |
| .        | .         | .                 | .                            |

Note that the modulus of this generator is 32, however we have realized a period of only 4, that is the sequence of 1, 9, 17, 25, 1, 9, . . . repeats after only 4 numbers. Obviously, care must be taken to insure that such occurrences do not happen in a random number generator. Hopefully, the period will also be independent of the starting value,  $X_0$ .

A great deal of work has been done on number theoretic considerations for the choice of  $m$  so as to yield a maximum period length (see Knuth<sup>3</sup>). To summarize, generators with modulus  $m = 2^p$  for any integer,  $p$ , can have a maximum period of  $m/4$ , or, for the System/360,  $2^{31}/4 = 2^{29}$ ; the period may also depend on the starting value. When the modulus  $m$  is a prime number, the maximum possible period is  $m - 1$ .

It so happens that the largest prime less than or equal to  $2^{31}$  is  $2^{31} - 1$ , which is most fortuitous. Hence, choosing  $m = 2^{31} - 1$  we can achieve a maximum period of  $m - 1 = 2^{31} - 2$ . These results produce only upper bounds on the period length. Recall in the example above, the maximum period possible is  $2^5/4 = 2^3 = 8$ , but that a period of only 4 was observed. This naturally leads to considerations of the choice of the multiplier,  $A$ .

Success in achieving a maximum period lies with the choice of the multiplier. Again, to briefly summarize the pertinent number theory, for a modulus  $2^{31}$  the multiplier  $A$  must differ by 3 from the nearest multiple of 8; the starting value,  $X_0$ , must be odd;  $A$  must be one greater than a multiple of 4; and  $C$  must be odd. These conditions only assure a maximum period of  $m/4$ , not necessarily good statistical properties. For the random number generator described here (LLRANDOM) we are choosing  $C = 0$ ; hence, this length is not achievable if  $m = 2^{31}$ . Luckily, the conditions on choosing  $A$  for the modulus  $m = 2^{31} - 1$  are more easily met and we can achieve the maximum period.

Utilizing some of the nice number theoretic properties of the number  $2^{31} - 1$ , to achieve a maximum period,  $A$  must be a positive primitive root of  $2^{31} - 1$  or a power of such a number. This is generally not easy to find; the value of  $A$  used in the generator described here is  $7^5$ . The number 7 is a positive primitive root of  $2^{31} - 1$  and raising 7 to the fifth power results in the multiplier 16807 which is also a positive primitive root of  $2^{31} - 1$  (Lewis, Goodman, and Miller<sup>1</sup>) and satisfies some conditions regarding the statistical performance of the generated sequence. These conditions will not be discussed here.

The generator

$$X_n \equiv 7^5 \cdot X_{n-1} \pmod{2^{31}-1} \quad (3)$$

is the generator reported in Lewis, Goodman, and Miller<sup>1</sup>. The authors cite the results of very extensive tests on this generator, all of which show that it is very satisfactory.

A. Division simulation. A practical consideration for random number generators is that they be fast, hopefully without requiring excessive memory to achieve speed. In many applications rather large quantities of numbers are needed and the speed of the generator can be crucial.

Nearly all random number generators are coded as subroutine or function subprograms in the assembler or machine language of the computer. The algorithm for implementing (3) is rather simple, involving a multiplication and then a division to effect the modulo operation. On most computers the division operation is rather slow as compared to the multiplication operation. In the past, the multiplier  $A$  was chosen so that its binary representation contained many zeroes, thereby speeding the multiplication. Unfortunately, this choice was at the expense of the period length, since such multipliers rarely met the number theoretic conditions for a maximum period. For the LLRANDOM generator (3) described here, the division operation has been replaced by a division simulation involving two shifts and an add instruction. Should a fixed-point (integer) overflow occur, two more additions are required to correct the situation.

The ordinary division on a System/360 Model 67-2 requires 8.49 micro-seconds. Without overflow, the simulation requires only 3.45

micro-seconds. When overflow occurs, the simulation takes an additional 2.32 micro-seconds for a total of 5.77 micro-seconds. These overflows occur quite rarely, on the order of only once in 250,000 iterations.

The division simulation algorithm (again due to Lehmer) is discussed by Payne, Rabung, and Bogyo<sup>2</sup> and works as follows. Define a congruence relationship by

$$X'_n \equiv A \cdot X_{n-1} \pmod{2^{31}}. \quad (4)$$

Performing the modulo operation on the product  $AX_{n-1}$  would give

$$AX_{n-1} = q2^{31} + r, \quad (5)$$

where  $q$  is some quotient and  $r$  is the remainder and is strictly less than  $2^{31}$ . Adding  $q$  to both sides of (4) we get

$$X_n = X'_n + q \equiv AX_{n-1} \pmod{2^{31}-1}. \quad (6)$$

This form gives the desired modulus of  $2^{31} - 1$ , if there is no overflow in the addition of  $X'_n + q$ . If there is overflow, to correct it we merely add a constant of 1 to get

$$X_n \equiv X'_n + q + 1 \pmod{2^{31}} = A \cdot X_{n-1} \pmod{2^{31}-1}, \quad (7)$$

which is again, the desired result.

This division simulation algorithm is very easily implemented on the System/360 and saves considerable execution time over conventional division.

B. Shuffling. The sequence produced by the generator (3) does appear to consist of independent, uniformly distributed numbers for most purposes.

We realize that the numbers are not actually independent, due to the procedure used to generate 'nem. It has been proposed that a sequence of such numbers be further randomized, or "shuffled," to improve upon the appearance of randomness (see, for instance, Knuth<sup>3</sup>). Serial correlation tests are usually employed to detect lack of independence in a sequence and at least one generator, RANDU, known to perform badly in a three-dimensional serial test was improved by shuffling. These tests will be discussed elsewhere. The various shuffling procedures which have been put forward have had little empirical validation.

The package described here has a built-in shuffling mechanism and it works as follows. A table of 128 random integers is maintained in the package. The starting values in the table represent members of the sequence (3) lagged by one million integers starting with an arbitrary seed. When a new integer is generated by the algorithm, its right-most seven bits are masked-off to form an index into the table ( $2^7=128$ ). The integer in the table indexed by the right-most seven bits is returned to the caller and that table entry is replaced by the integer just generated. In essence, we are taking "chunks" of 128 numbers from the basic sequence and shuffling them before they are used.

This particular shuffling scheme is dependent on the choice of the modulus. For a modulus of  $2^{31}$  the right-most bits of a congruential random number generator are non-random and their use in this scheme would defeat the purpose of shuffling. However, with a modulus of  $2^{31} - 1$  and the positive primitive root multiplier  $A = 7^5$ , the right-most bits are quite random and the desired results are obtained.

C. Uniform (0.0,1.0) random numbers. So far, we have discussed how to generate uniform random integers over the range 1 to  $m = 2^{31} - 1$ . In most applications, uniform random numbers over the range 0.0 to 1.0 are desired. In theory, the uniform integers,  $X_i$ , are divided by  $m$  to produce these numbers, as

$$U_i = X_i/m. \quad (8)$$

In actual implementation on the System/360, the integer result is algebraically shifted right seven bits and a normalized floating point exponent is logically OR'ed on to it. The result is a properly normalized floating point random number over the range 0.0 to 1.0, usually referred to as a "real" uniform number.

D. Normally distributed random deviates. The uniformly distributed random numbers described above are not only useful in their own right, but form the basis of transformations into random numbers with other probability distributions. One of the most important of these distributions is the Normal distribution.

There are several methods of approximating a Normal distribution with uniform random numbers. One of the oldest and, unfortunately, most common is the "sum of  $k$  uniforms method." The algorithm is based on the fact that the uniform (0.0,1.0) distribution has a mean of  $1/2$  and a standard deviation of  $\sqrt{1/12}$ . The algorithm works as follows:

$$X = \frac{\sum_{i=1}^k U_i - k/2}{\sqrt{k/12.0}} \quad (9)$$

The random deviate  $X$  is approximately normally distributed with mean 0 and variance 1. The approximation is not as good as other methods and it is rather time consuming in that  $k$  uniforms must be generated and then summed. It was basically devised to overcome the very time consuming multiply and divide operations in older computers.

A more accurate algorithm is known as the Box-Muller method or Polar method which is actually a rejection method due to von Neumann. The method requires the generation of two uniforms to produce two independent Normals. It is based on the distribution of points inside the unit circle. The method is more accurate than the "sum of  $k$  uniforms method" (in fact, theoretically perfect). However, it does require two square roots and two natural logarithm operations which are generally rather time consuming.

The algorithm used in the package described here is based on a method developed by Marsaglia and is known as the "rectangle-wedge-tail" method. This algorithm is by far the fastest algorithm available for generating normally distributed random numbers, although it requires more memory than the Polar method.

The second volume of Knuth's "The Art of Computer Programming"<sup>3</sup> gives a complete and detailed description of the algorithm. Briefly, the positive half of the Normal density curve is discretized into 37 rectangles, wedges, and a tail as in Figure 1. All of the rectangles are uniformly distributed densities. The wedges are approximated by "nearly linear densities." Finally, the tail distribution is computed by a modification to the Polar method. The normal density,  $f(x)$ , is then given by the composite function.

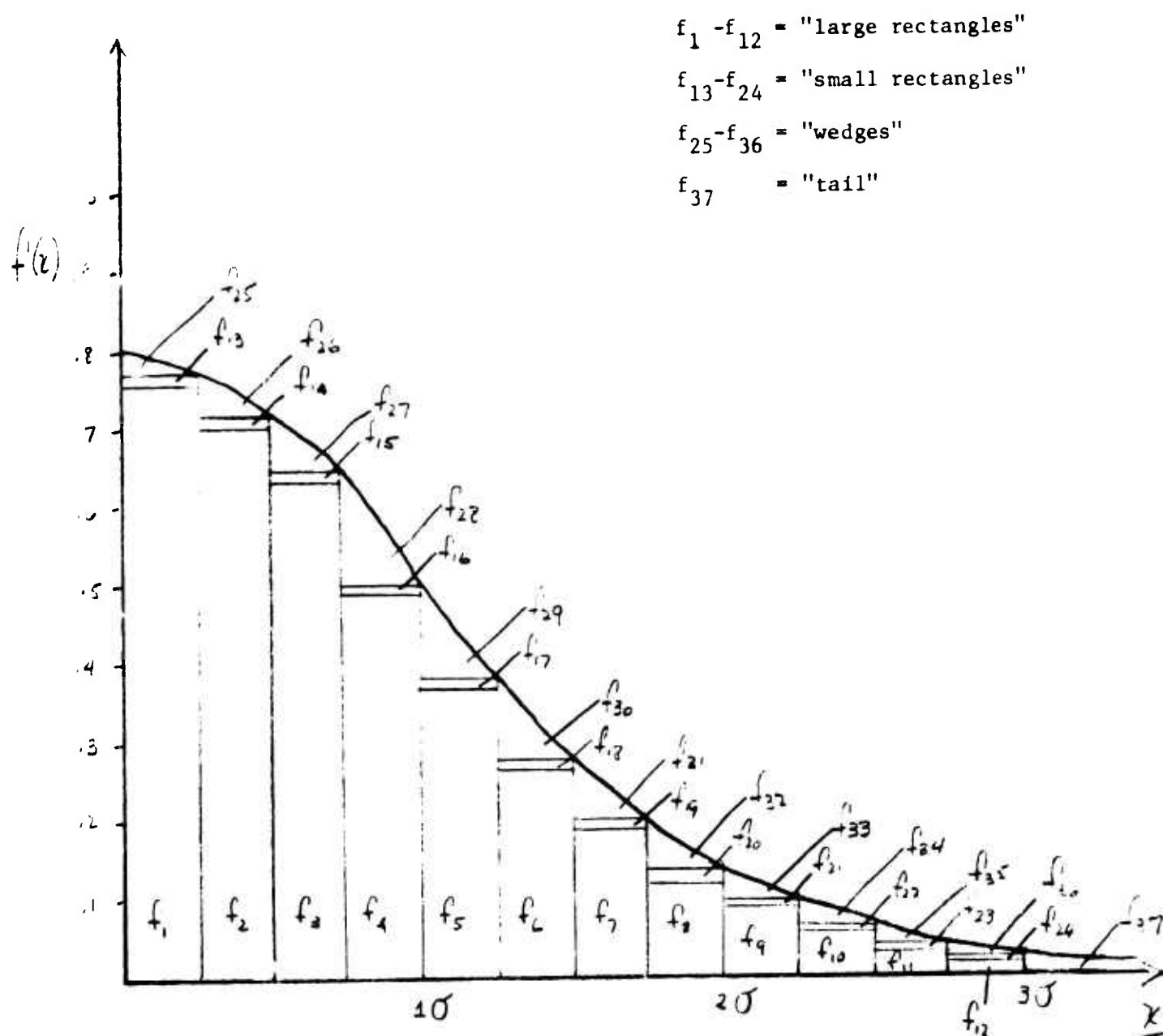


FIGURE 1

RECTANGLE-WEDGE-TAIL METHOD OF APPROXIMATING THE NORMAL DENSITY

$$f(x) = p_1 f_1(x) + p_2 f_2(x) + \dots + p_{37} f_{37}(x), \quad (10)$$

where

$$\sum_{i=1}^{37} p_i = 1,$$

the densities  $f_1$  to  $f_{24}$  are the rectangles;  $f_{25}$  to  $f_{36}$  are the wedges; and  $f_{37}$  is the tail. The first twelve uniformly distributed rectangles are used 88% of the time. This makes for an extremely fast algorithm for the majority of deviates. When the tail is sampled, the deviate is generated by a modified Polar method and still quite satisfactory.

This generator for Normal deviates, like nearly all others, produces deviates with zero mean and unit variance. To change the scale and shape to any mean,  $\mu$ , and the standard deviation,  $\sigma$ , we apply the linear transformation

$$Z = \mu + \sigma X \quad (11)$$

where  $z$  now has the desired shape and scale parameters.

E. Exponential distribution. Another probability distribution of major interest in simulations is the exponential. The cumulative distribution function and probability density function for the exponential are respectively

$$F(x) = 1 - e^{-\lambda x}, \quad (12)$$

$$f(x) = \lambda e^{-\lambda x}. \quad (13)$$

The expected value of the exponential distribution is:

$$E[X] = 1/\lambda. \quad (14)$$

The problem of generating exponential deviates reduces to one of generating "unit" exponentials, i.e. those with  $\lambda = 1$ , and then multiplying the result by whichever  $\lambda$  is necessary to give the desired distribution.

One of the most common methods of generating numbers from distributions other than the uniform is to use the inverse transformation technique (see Gaver and Thompson<sup>4</sup>). This can be described graphically, as in Figure 2, with a plot of the distribution function

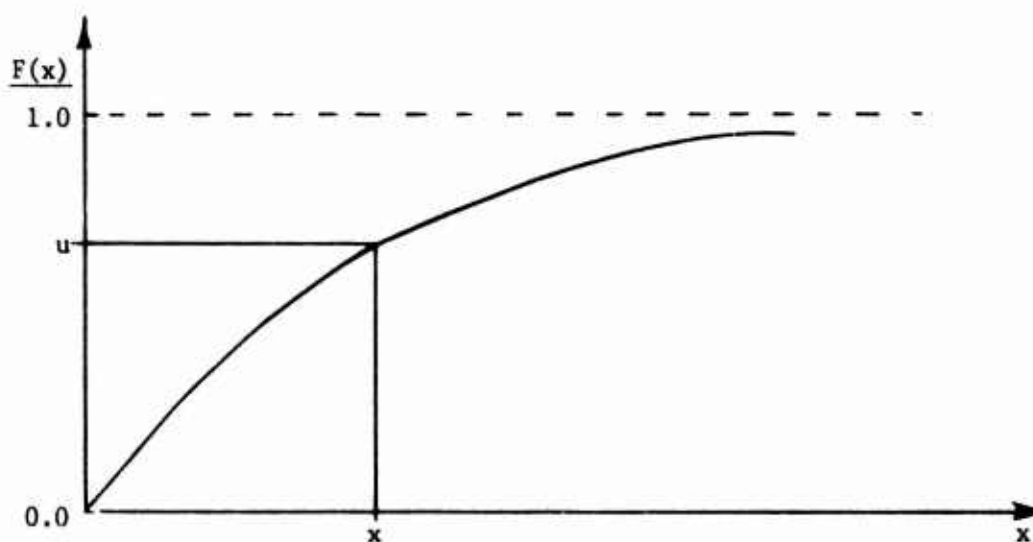


FIGURE 2.

#### CUMULATIVE DISTRIBUTION FUNCTION OF THE EXPONENTIAL DISTRIBUTION

The range of the abscissa,  $X$ , is infinite in extent. However, the range of the ordinate,  $F(x)$ , is  $(0.0, 1.0)$ , the range of uniform  $(0,1)$  random variables. The inverse transformation technique is to generate a uniform random number, say  $U$ , and use this as the ordinate. The exponential deviate, say  $X$ , is the abscissa point corresponding to the intersection of the ordinate and the curve.

Mathematically, this technique is expressed as

$$\begin{aligned} u &= F(x) = 1 - e^{-x}, & \lambda &= 1, \\ x &= F^{-1}(u), \end{aligned} \tag{15}$$

where  $u$  is the uniform random number. This inverse transformation is rather easily implemented for exponentially distributed random variables via natural logarithms since we get

$$F^{-1}(U) = -\ln(1-U),$$

or by the symmetry of the uniform distribution

$$F^{-1}(U) = -\ln(U).$$

Perhaps the most common implementation of exponential deviate generators is this natural logarithm transformation. It is mathematically appealing as well as trivial to program, given the usual FORTRAN subroutines.

The exponential deviate generator in the LLRANDOM package is based on Marsaglia's method of dividing the probability density into a series of rectangles, wedges, and a tail. Although more complicated to program and larger in size, this method is approximately 40% faster than the logarithmic transformation.

For a survey of the generation of normal and exponentially distributed variables see Ahrens and Dieter<sup>5</sup>.

### III. HOW TO USE THE PACKAGE.

The random number package described here is intended solely for use on the IBM System/360 or System/370 computers. The package consists of one Assembler F control section (CSECT) with nine entry points and two FORTRAN IV function subprograms. The names of the entry points and their functions are summarized in Table 1.

The subroutine entry point OVFLOW has no calling arguments and should be called once and only once at the beginning of the user's main FORTRAN program. The function subprograms RNORTH and REXPTH are called by the Assembler routine as needed and should not be called by the user. The eight additional entry points are the names of the actual routines to generate the random numbers. There are four types of random numbers which can be generated:

- (1) uniformly distributed integers on the range 1 to  $2^{31} - 1$ ;
- (2) uniformly distributed single precision floating point numbers between 0.0 and 1.0;
- (3) single precision floating point normal deviates with mean zero and variance 1; and
- (4) single precision floating point exponential deviates with mean 1.

There is a separate entry point for each of the four types if shuffling of the sequence is desired.

For all eight entry points, the FORTRAN calling sequence is the same, namely:

CALL (entry point) (IX, A, N)

where

(entry point) refers to the routine desired,

viz. INT, SINT, RANDOM, SRAND, NORMAL, SNORM, EXPON, or SEXPON;

| ENTRY<br>POINT | FUNCTION                                                                                  |
|----------------|-------------------------------------------------------------------------------------------|
| OVFLOW         | Calls SPIE, handles fixed point overflows. (Must be called once at start of program.)     |
| INT            | Generates integer random numbers.                                                         |
| SINT           | Generates shuffled integer random numbers.                                                |
| RANDOM         | Generates single precision floating point (0.0,1.0) random numbers.                       |
| SRAND          | Generates single precision floating point (0.0,1.0) shuffled random numbers.              |
| NORMAL         | Generates single precision floating point normal deviates ( $\mu=0, \sigma=1$ ).          |
| SNORM          | Generates shuffled single precision floating point normal deviates ( $\mu=0, \sigma=1$ ). |
| EXPON          | Generates single precision floating point exponential deviates ( $\lambda=1$ ).           |
| SEXPON         | Generates shuffled single precision floating point exponential deviates ( $\lambda=1$ ).  |

TABLE 1.

ENTRY POINT NAMES OF CONTROL SECTION OVFLOW

- IX is the starting value of the sequence and may contain any integer number between 1 and 2147483647. This variable should not be altered by the user during the execution of the program, unless it is desired to repeat a sequence of random numbers.
- A is either a scalar or vector variable and is the location with a specified dimension into which the random number or numbers are stored (see next parameter). Note that for entry points INT and SINT, this argument should be of INTEGER type.
- N is an integer variable or constant designating how many random numbers are to be generated during this call. If N is greater than 1, A above must be a vector dimensioned at least as large as N. If N is equal to 1, then A may be scalar.

Some sample programs are given below:

- (1) To generate 1000 consecutive integer random numbers:

```

INTEGER*4  M(1000)

CALL OVFLOW

IX = 1234567

--

--

--

CALL INT (IX, M, 1000)

--

END

```

- (2) To generate 25 shuffled single precision floating point normal deviates and scale to mean 10 and standard deviation 5:

```

REAL*4  A(25)

CALL OVFLOW

JJ = 1936748

N = 25

--

--

--

```

```

CALL SNORM (JJ, A, N)

DO 1 I = 1,25

A(I) = A(I)*5.0 + 10.0

1 CONTINUE

```

```

--

```

```

--

```

```

--

```

```

END

```

- (3) To generate one single precision floating point exponential deviate with mean 6:

```

CALL OVFLOW

```

```

I9 = 98367221

```

```

--

```

```

--

```

```

--

```

```

CALL EXPON (I9, E, 1)

```

```

E = E*6.0

```

```

--

```

```

--

```

```

--

```

```

END

```

A. Implementation. LLRANDOM was designed and coded to run under Operating System/360 (OS). The Assembler Language control section contains a SPIE (Set Program Interrupt Exit) macro instruction which is a part of the OS Supervisor Services. This macro enables LLRANDOM to correct for the fixed point overflows resulting from the division simulation algorithm.

The remainder of the assembly coding is in Basic Assembler Language (BAL), i.e. no other macro calls or supervisor calls. To run LLRANDOM under another operating system for the System/360, an appropriate substitution for the SPIE macro would be necessary.

As currently programmed, LLRANDOM has the following memory requirements:

| <u>MODULE</u>            | <u>SIZE IN BYTES<br/>(DECIMAL)</u> |
|--------------------------|------------------------------------|
| Assembler CSECT          | 3571                               |
| FORTTRAN function RNORTH | 1512                               |
| FORTTRAN function REXPTH | <u>1106</u>                        |
| Total memory requirement | <u>6189</u>                        |

The System/360 internal timer is rather crude for timing the execution of programs. The following times are therefore approximate timings for the generation of pseudo-random numbers on a System/360 Model 67-2.

| <u>ENTRY POINT</u> | <u>TIME IN MICROSECONDS</u> |                                           |
|--------------------|-----------------------------|-------------------------------------------|
| INT                | 10.7                        |                                           |
| SINT               | 15.7                        |                                           |
| RANDOM             | 15.6                        |                                           |
| SRAND              | 20.0                        |                                           |
| NORMAL             | 57.5                        | (Polar method takes 349 microseconds)     |
| SNORM              | 65.8                        |                                           |
| EXPON              | 59.1                        | (Logarithm method takes 132 microseconds) |
| SEXPON             | 68.4                        |                                           |

B. Future Enhancements. The normal and exponential deviate routines in LLRANDOM are patterned after a package, SUPER-DUPER, available from Professor G. Marsaglia at McGill University in Montreal. It is available at the Naval Postgraduate School. Marsaglia uses a different multiplier,  $A$ , and modulus,  $m$ , in his congruential generator from that used in LLRANDOM and he then exclusive OR's this result with the output of a feedback shift register generator. SUPER-DUPER provides only one deviate per call and does not provide for shuffling the sequence.

The two FORTRAN function subprograms, RNORTH and REXPTH, are taken (with slight modification) directly from SUPER-DUPER. Among the changes to be made to LLRANDOM will be to rewrite RNORTH and REXPTH in System/360 Assembly Language and incorporate them directly into the package.

We have experienced occasions where large-scale simulations have been coded in FORTRAN using double precision variables. The fact that LLRANDOM returns single precision numbers causes some inconvenience. To alleviate this problem we will provide the capability in LLRANDOM to return single precision numbers into double precision variables or arrays. Note that the values returned will still be single precision; however, they will be stored properly into double precision locations.

Finally, additional entry points will be added to provide single precision floating point gamma deviates. Shuffling of the gamma deviates will also be available.

Other enhancements are under consideration.

REFERENCES

1. Lewis, P. A. W., A. S. Goodman and J. M. Miller, Pseudo-Random Number Generator for the System/360, IBM Systems Journal, No. 2, 1969.
2. Payne, W. H., J. R. Rabung and T. P. Bogyo, Coding the Lehmer Pseudo-random Number Generator, Communications of the ACM, Vol. 12, No. 2, February 1969.
3. Knuth, D. E., The Art of Computer Programming, Volume 2: Seminumerical Algorithms, Addison-Wesley, Reading, Mass., 1969.
4. Gaver, D. P. and G. L. Thompson, Programming and Probability Models in Operations Research, Brooks/Cole, Monterey, Calif., 1973.
5. Ahrens, J. H. and U. Dieter, Computer methods for sampling from the exponential and normal distributions, Comm. of the ACM, 15, 873-882, 1972.
6. Lewis, P. A. W., Large-Scale Computer-Aided Statistical Mathematics, Proceedings of the Computer Science and Statistics 6th Annual Symposium on the Interface, Michael E. Tarter (ed.), University of California, Berkeley, 1973.

\*\*\*\*\* NAVAL POSTGRADUATE SCHOOL RANDOM NUMBER GENERATOR : LLRANDOM \*\*\*\*\*

```
C      REXP TOOTH FUNCTION  
FUNCTION REXPTH(K,IX)  
DIMENSION C(65)  
DATA C/Z40F0000,Z40E10000,Z40D40000,Z40C70000,Z40B80000,  
Z40A90000,Z40A50000,Z40980000,Z40910000,Z40890000,Z40800000,  
Z40780000,Z40710000,Z406A0000,Z40640000,Z405E0000,Z40580000,  
Z40530000,Z404E0000,Z40490000,Z40440000,Z40400000,Z403C0000,  
Z40390000,Z40350000,Z40320000,Z402F0000,Z402C0000,Z40290000,  
Z40270000,Z40240000,Z40220000,Z40200000,Z401E0000,Z401C0000,  
Z401A0000,Z40190000,Z40170000,Z40160000,Z40150000,Z40130000,  
Z40120000,Z40110000,Z40100000,Z3FA00000,Z3F9000000,  
Z3FC00000,Z3FB00000,Z3F800000,Z3F700000,Z3F600000,Z3F600000,  
Z3F800000,Z3F800000,Z3F700000,Z3F700000,Z3F600000,Z3F600000,  
Z3F6C0000,Z3F500000,Z3F500000,Z3F400000,Z3F400000,Z3F400000/  
DATA I1/ZFB4FAA9I/  
IF(K.GT.I1)GO TO 5  
CALL RANDOM(IX,I1)  
IF(U1.GT.:7917049) GO TO 3  
Y=1.-I.235962*U1  
REXPTH=-ALOG(T)  
J=16.*REXPTH+1.  
CALL RANDOM(IX,Z9,I1)  
IF(Z9*(.0604*T+.0039).GT.T-C(J)) GO TO 1  
RETURN  
REXPTH=19.20352*U1-15.20352  
J=16.*REXPTH+1.  
EX=EXP(-REXPTH)  
CALL RANDOM(IX,Z9,I1)  
IF(Z9*(.0604*EX+.0039).GT.EX-C(J)) GO TO 1  
RETURN  
CALL RANDOM(IX,U1,I1)  
IF(U1.EQ.0)GO TO 5  
REXPTH=4.-ALOG(U1)  
RETURN  
END
```

\*\*\*\*\* NAVAL POSTGRADUATE SCHOOL RANDOM NUMBER GENERATOR : LLRANDOM \*\*\*\*\*

```

C RNCR TOOTH FUNCTION
FUNCTION RNORTH(K,IX)
DIMENSION C(45)
DATA C/Z40FD2B5F,Z40FD2B5F,Z40FAA9AD,Z40F5A648,Z40F32496,
$ Z40EE2131,Z40E69C1A,Z40E19885,Z40DA139E,Z40D28E87,Z40C887BE,
$ Z40C102A6,Z40B6FBDD,Z40ACF513,Z40A2EE4A,Z4098E78C,Z40916269,
$ Z408758AC,Z407D54D6,Z40734E0D,Z406BC8F6,Z4061C22C,Z405A3D15,
$ Z405287FE,Z404832E7,Z4043ADD0,Z403C2889,Z40372554,Z402FA03D,
$ Z402A9CD8,Z40259973,Z4020960E,Z401E145C,Z401910F7,Z40168F45,
$ Z40140D93,Z40118BE0,Z3FF0A2E4,Z3FC887BE,Z3FA06C98,Z3F785172,
$ Z3F785172,Z3F50364C,Z3F50364C/
DATA I1/ZF8C35400/I2/ZFE79702E/
IF(K.GT.I1)GO TO 3
CALL RANDOM(IX,S,I)
CALL RANDCM(IX,T,I)
B=AIN(T,*(S+T))+37.*ABS(S-T)
CALL RANDOM(IX,Z9,I)
CALL RANDCM(IX,Z8,I)
X=Z9-Z8
RNORTH=.0625*(X+SIGN(B,X))
RETURN
IF(K.GT.I2)GO TO 5
CALL RANDCM(IX,Z9,I)
CALL RANDCM(IX,Z8,I)
IF(Z8.GT.0.50) Z9=-Z9
RNORTH=2.75*Z9
J=16.*ABS(RNORTH)+1.
IF(J-14) 6,6,7
P=(J+J-1)*.1497466E-2
GO TO 8
P=(89-J-J)*.698817E-3
CALL RANDCM(IX,Z9,I)
IF(Z9.GT.79.78846*(EXP(-.5*RNORTH*RNORTH)
$ -C(J)-P*(J-16.*ABS(RNORTH)))) GOTO 4
RETURN
CALL RANDCM(IX,V,I)
CALL RANDCM(IX,Z9,I)
IF(Z9.GT.0.5) V=-V
IF(V.EQ.0) GO TO 5
X=SQR(7.5625-2.*ALOG(ABS(V)))
CALL RANDCM(IX,Z9,I)
IF(Z9*X.GT.2.75) GO TO 5
RNORTH=SIGN(X,V)
RETURN
END

```

3 4

6

7

8

5

\*\*\*\*\* NAVAL POSTGRADUATE SCHOOL RANDOM NUMBER GENERATOR : LLRANDOM \*\*\*\*\*

```

OVFLOW      CSECT      RNRTH,RXPTH
EXTRN       INT,SINT,RANDOM,SRAND,NORMAL,SNORM,EXPON,SEXPON
ENTRY       OVFLOW,R12
USING       B(1,R15)      BRANCH AROUND ID
            AL1(6)
            CL6,OVFLOW,
            R14,R12,12(R13)  SAVE REGISTERS IN HIGH SAVE AREA
            R12,R15        ESTABLISH BASE ADDRESS
            R13,SA+4       SAVE CALLER'S R13
            R2,R13
            R13,SA         NEW SAVE AREA
            R13,8(R2)      STORE WITH CALLING ROUTINE

            ISSUE SPIE TO GET FIXED POINT OVERFLOWS AS WELL AS FORTRAN
            INTERRUPTS.

            SPIE          FIXIT,(8,9,12,13,15)
            ST            R1,PICA          SAVE FORTRAN'S PICA ADDRESS
            LM            R13,SA+4        RESTORE CALLER'S R13
            BCR           R14,R12,12(R13) RESTORE THE REGISTERS
            BCR           15,R14         RETURN

```

\*\*\*\*\* NAVAL POSTGRADUATE SCHOOL RANDOM NUMBER GENERATOR : LLRANDOM \*\*\*\*\*

\* \* \* SPIE BRINGS US HERE ON INTERRUPTS

FIXIT

```

      USING *,R15,X*F7'      WAS IT A FIXED PCINT OVERFLOW? IT
      TM 7(R1),X*F7'      NO, LET FORTRAN'S SPIE HANDLE IT
      BC 5,FORT      LET FORTRAN'S SPIE HANDLE INTERRUPTED
      CLC 17(3,R1),AINT+1      TEST WHETHER BASE OF ENTRIES INT AND
      BL 0(,R14)      ROUTINE WAS BETWEEN ENTRIES INT AND
      CLC 17(3,R1),ASEXPO+1      SEXPON INCLUSIVE; IF NOT, IGNORE
      BH 0(,R14)      THE INTERRUPT
      A ADD 2**31-3
      AR 4,MORE TO MAKE 2**31+1 CORRECTION
      BR R4,R2      ADD 4 MORE TO CONTINUE
      L R14      ALL FIXED, POINT OVERFLOW; LET FORTRAN
      FR R15,0(,R15)      NOT EXTENDED ERROR HANDLING RCUTINE
      R15      EXTENDED ERROR OF IT

```

LLRA0240  
LLRA0250  
LLRA0260  
LLRA0270  
LLRA0280  
LLRA0290  
LLRA0300  
LLRA0310  
LLRA0320  
LLRA0330  
LLRA0340  
LLRA0350  
LLRA0360  
LLRA0370  
LLRA0380  
LLRA0390

\*\*\*\*\* NAVAL POSTGRADUATE SCHOOL RANDOM NUMBER GENERATOR : LLRANDOM \*\*\*\*\*

```

* * *
ENTRY POINT : INT
INT
  CNUP 0,8
  USING INT,R15
  B DC 8(R15)
  DC AL1(3)
  DC CL3,INT'
  STM R14,R12,12(R13)
  ST R13,SA+4
  LR R2,R13
  LA R13,SA
  ST R13,8(R2)
  L R9,A75
  LA R2,4
  LM R5,R7,0(R1)
  L R5,0(R5)
  L R3,0(R7)
  L R3,2
  SLA R6,R2
  SR R7,R2
  L R7,8
  CNRJP R4,R9
  MRDA R4,1
  SRL R5,1
  AR R4,R5
  ST R5,R4
  ST R5,0(R7,R6)
  BXLE R7,R2,11
  L R4,0(R1)
  ST R5,0(R4)
  L R13,SA+4
  LM R14,R12,12(R13)
  BCR 15,R14

  BASE REGISTER
  BRANCH AROUND ID

  SAVE REGISTERS IN HIGH SAVE AREA
  ADDRESS OF HIGH SAVE AREA IN LOW SAVE AR.
  COPY TO R2
  ADDRESS OF LOW SAVE AREA
  LOAD MULTIPLIER
  CONSTANT FOR BXLE
  ADDRESSES OF THREE ARGUMENTS
  LOAD STARTING VALUE INTO R5
  NUMBER OF CONSECUTIVE WORDS TO FILL
  CONVERT TO BYTES
  BACKUP ONE WORD IN CALLER'S ARRAY
  INITIAL VALUE FOR INDEX REGISTER
  ALIGN BXLE LOOP FOR SPEED
  FORM PRODUCT OF A AND X(N-1)
  R4 = REMAINDER : TO REMAINDER BY 2**31-1
  ADD QUOTIENT TO DIVISION BY 2**31-1
  PUT X(N) INTO R5 FOR NEXT GC AROUND
  STORE IN CALLER'S ARRAY
  LOOP AROUND AGAIN
  GET STARTING VALUE ADDRESS AGAIN
  STORE AS STARTING VALUE FOR NEXT CALL
  RESTORE CALLER'S SAVE AREA POINTER
  RESTORE THE REGISTERS
  RETURN

```

LLRA0410  
 LLRA0420  
 LLRA0430  
 LLRA0440  
 LLRA0450  
 LLRA0460  
 LLRA0470  
 LLRA0480  
 LLRA0490  
 LLRA0500  
 LLRA0510  
 LLRA0520  
 LLRA0530  
 LLRA0540  
 LLRA0550  
 LLRA0560  
 LLRA0570  
 LLRA0580  
 LLRA0590  
 LLRA0600  
 LLRA0610  
 LLRA0620  
 LLRA0630  
 LLRA0640  
 LLRA0650  
 LLRA0660  
 LLRA0670  
 LLRA0680  
 LLRA0690  
 LLRA0700  
 LLRA0710  
 LLRA0720  
 LLRA0730  
 LLRA0740

\*\*\*\*\* NAVAL POSTGRADUATE SCHOOL RANDOM NUMBER GENERATOR : LLRANDOM \*\*\*\*\*

```

* * * ENTRY POINT : SINT
* * *
SINT
  CNOP 0,8
  USING SINT,R15
  B 10(C,R15)
  DC AL1(4)
  DCM CL4,SINT
  STM R14,R12,12(R13)
  ST R13,SA+4
  LR R2,R13
  LA R13,SA
  ST R13,8(R2)
  LA R9,A75
  LM R2,4
  LA R5,7,0(R1)
  LM R5,0(,R5)
  LA R3,0(,R7)
  LA R3,22
  SR R6,R2
  LR R7,R2
  LA R8,TABLE
  LA R1,MASK
  L CNOP 0,8
  CMR R4,R9
  SLDA R4,1
  SRL R5,1
  AR R5,R4
  LNR R4,R1
  SLA R4,2
  R0,0(R4,R8)
  ST R5,0(R4,R8)
  ST R0,0(R7,R6)
  ST R7,R2,L2
  BXLE R13,SA+4
  L L R1,24(,R13)
  L L R4,0(,R1)
  L L R5,0(,R4)
  L L R14,R12,12(R13)
  L L R15,R14
  BCR
  BASE REGISTER
  BRANCH AROUND ID
  SAVE REGISTERS IN HIGH SAVE AREA
  ADDRESS OF HIGH SAVE AREA IN LOW SAVE AR.
  COPY TO R2
  ADDRESS OF LOW SAVE AREA
  LOAD MULTPLIER
  CONSTANT FOR BXLE
  ADDRESSES OF THREE ARGUMENTS
  LOAD STARTING VALUE INTO R5
  NUMBER OF CONSECUTIVE WORDS TO FILL
  CONVERT TO BYTES IN CALLER'S ARRAY
  BACKUP ONE WORD FOR INDEX REGISTER
  INITIAL VALUE OF SHUFFLING TABLE
  ADDRESS OF SHUFFLING TABLE
  INDEX MASK FOR SHUFFLING
  ALIGN BXLE LOOP FOR SPEED
  FORM PRODUCT OF A AND X(N-1)
  R4 = REMAINDER ; TO REMAINDER THEREBY
  PUT X(N) INTO R5 FOR NEXT GC AROUND
  SIMULATING DIVISION BY 2**31-1
  EXTRACT RIGHT-MOST 7 BITS
  CONVERT TO BYTE OFFSET IN TABLE
  SELECT RANDOM TABLE VALUE
  REPLACE TABLE VALUE WITH X(N)
  RANDOM TABLE VALUE TO CALLER'S ARRAY
  LOOP AROUND AGAIN
  RESTORE CALLER'S SAVE AREA POINTER
  GET ARGUMENT LIST POINTER AGAIN
  STORE STARTING VALUE ADDRESS AGAIN
  STORE AS STARTING VALUE FOR NEXT CALL
  RESTORE THE REGISTERS
  RETURN

```

L2

N2

LLRA0760  
 LLRA0770  
 LLRA0780  
 LLRA0790  
 LLRA0800  
 LLRA0810  
 LLRA0820  
 LLRA0830  
 LLRA0840  
 LLRA0850  
 LLRA0860  
 LLRA0870  
 LLRA0880  
 LLRA0890  
 LLRA0900  
 LLRA0910  
 LLRA0920  
 LLRA0930  
 LLRA0940  
 LLRA0950  
 LLRA0960  
 LLRA0970  
 LLRA0980  
 LLRA0990  
 LLRA1000  
 LLRA1010  
 LLRA1020  
 LLRA1030  
 LLRA1040  
 LLRA1050  
 LLRA1060  
 LLRA1070  
 LLRA1080  
 LLRA1090  
 LLRA1100  
 LLRA1110  
 LLRA1120  
 LLRA1130  
 LLRA1140  
 LLRA1150  
 LLRA1160

[illegible]



\*\*\*\*\* NAVAL POSTGRADUATE SCHOOL RANDOM NUMBER GENERATOR : LLRANDOM \*\*\*\*\*

LLRA2130  
LLRA2140  
LLRA2150

ADD ZERO AND NORMALIZE  
STORE BACK NORMALIZED  
CONTINUE THE BXLE LOOP

FR2,FR0  
FR2,C(R7,R6)  
R12

AER  
STE  
BR

\*\*\*\*\* NAVAL POSTGRADUATE SCHOOL RANDOM NUMBER GENERATOR : LLRANDOM \*\*\*\*\*

```

*      ENTRY PCINT :  NORMAL
*
*      NORMAL
*
*      CNOP 0,8
*      USING 12(,R15)
*      DC 12(,R15)
*      DC AL1(6)
*      DC CL6,NORMAL'
*      STM R14,R12,12(R13)
*      LTR R13,SA2+4
*      LST R2,R13
*      LST R13,SA2
*      LST R13,8(,R2)
*      LST R9,R11,A75N
*      LST R2,4
*      LST R5,R7,0(R1)
*      LST R5,0(,R5)
*      LST R3,0(,R7)
*      LST R3,2
*      LST R6,R2
*      LST R7,R2
*      LST R13,ATBLE
*      LST R12,N5
*      LST O,8
*      LST R4,R9
*      LST R4,1
*      LST R5,1
*      LST R5,R4
*      LST R0,R5
*      LST R4,R11
*      LST R8,F1
*      LST R5,R5
*      LST R4,R4
*      LST R5,C1
*      LST 11,F2
*      LST R4,8
*      LST R4,0(R4,R13)
*      LST R4,PWRD+1
*      LST R5,8
*      LST R5,R10
*      LST R5,0(R7,R6)
*      LST FRO,PWRD
*      LST FRO,0(R7,R6)
*      LST R5,R0
*      LST R12
*      LST R5,C2
*      LST 11,F3
*
*      BASE REGISTER
*      BRANCH AROUND ID
*
*      SAVE REGISTERS IN HIGH SAVE AREA
*      COPY TO R2
*      ADDRESS OF HIGH SAVE AREA IN LOW SAVE AR.
*      ADDRESS OF LOW SAVE AREA
*      ADDRESS OF LOW SAVE AREA IN HIGH SAVE AR.
*      LOAD MULTIPLIER, EXPONENT, AND TEST MASK
*      CONSTANTS FOR BXLE
*      ADDRESSES OF THREE ARGUMENTS
*      LOAD STARTING VALUE INTO R5
*      NUMBER OF CONSECUTIVE WORDS TO FILL
*      CONVERT TO BYTES
*      BACKUP ONE WORD IN CALLER'S ARRAY
*      INITIAL VALUE FOR INDEX REGISTER
*      ADDRESS OF TABLE OF CONSTANTS
*      ADDRESS OF BXLE
*      ALIGN BXLE LOOP FOR SPEED
*      FORM PRODUCT OF A AND X(N-1)
*      R4 = REMAINDER : R5 = QUOTIENT
*      ADD QUOTIENT TO REMAINDER THEREBY
*      PUT X(N) INTO R5
*      COPY R5 INTO R0 FOR NOW
*      SHOULD WE MAKE IT NEGATIVE ?
*      POSITIVE, KEEP GOING
*      MAKE R5 TRUE NEGATIVE
*      CLEAR R4 TO ZERO
*      R5 LESS THAN X'68000000' ?
*      NO
*      SHIFT FIRST 8 BITS OF R5 INTO R4 AS INDEX
*      OBTAIN CONSTANT FROM TABLE
*      STORE IN SECOND BYTE OF PWRD
*      SHIFT REMAINING 24 BITS RIGHT THEN OR ON
*      EXPONENT TO MAKE (24 BITS)/16
*      STORE IN CALLER'S ARRAY
*      LOAD CHARACTERISTIC TO FLOATING POINT
*      REGISTER 0 AND ADD FRACTION
*      STORE NORMAL DEViate IN CALLER'S ARRAY
*      COPY BACK TO R5 FOR NEXT GO AROUND
*      GO TO BXLE AND CONTINUE
*      R5 LESS THAN X'D0000000' ?
*      NO

```

LLRA2170  
 LLRA2180  
 LLRA2190  
 LLRA2200  
 LLRA2210  
 LLRA2220  
 LLRA2230  
 LLRA2240  
 LLRA2250  
 LLRA2260  
 LLRA2270  
 LLRA2280  
 LLRA2290  
 LLRA2300  
 LLRA2310  
 LLRA2320  
 LLRA2330  
 LLRA2340  
 LLRA2350  
 LLRA2360  
 LLRA2370  
 LLRA2380  
 LLRA2390  
 LLRA2400  
 LLRA2410  
 LLRA2420  
 LLRA2430  
 LLRA2440  
 LLRA2450  
 LLRA2460  
 LLRA2470  
 LLRA2480  
 LLRA2490  
 LLRA2500  
 LLRA2510  
 LLRA2520  
 LLRA2530  
 LLRA2540  
 LLRA2550  
 LLRA2560  
 LLRA2570  
 LLRA2580  
 LLRA2590  
 LLRA2600  
 LLRA2610  
 LLRA2620  
 LLRA2630  
 LLRA2640  
 LLRA2650

L5

F1

F2



\*\*\*\*\* NAVAL POSTGRADUATE SCHOOL RANDOM NUMBER GENERATOR : LLRANDOM \*\*\*\*\*

```

N5      L      R5,XWRD      NEW STARTING VALUE
      LA      R13,ATBLE     RESTORE R13 TO TABLE OF CONSTANTS
      BXLE    R7,R2,L5      LOOP AROUND AGAIN
      L       R13,SA2+4     RESTORE ARE HIGH SAVE AREA POINTER
      L       R1,24(,R13)   GET ARGUMENT LIST POINTER AGAIN
      L       R4,0(,R1)     GET STARTING VALUE ADDRESS AGAIN
      ST      R5,0(,R4)     STORE AS STARTING VALUE FOR NEXT CALL
      LM      R14,R12,L2(R13) RESTORE THE REGISTERS
      BCR     15,R14        RETURN

```

```

LLRA3150
LLRA3160
LLRA3170
LLRA3180
LLRA3190
LLRA3200
LLRA3210
LLRA3220
LLRA3230

```

\*\*\*\*\* NAVAL POSTGRADUATE SCHOOL RANDOM NUMBER GENERATOR : LLRANDOM \*\*\*\*\*

```

* * *
ENTRY POINT : SNORM
SNORM
  CNOP Q,8
  USING SNORM,R15
  B 10(,R15)
  DC AL1(5)
  DC CL5,SNORM,
  STM F14,R12,12(R13)
  ST R13,SA2+4
  LA R13,SA2
  ST R13,8(,R2)
  LM R9,R11,A75N
  LA R2,4
  LM R5,R7,0(R1)
  LM R5,0(,R5)
  LM R3,0(,R7)
  LA R3,2
  SR R6,R2
  LA R7,R2
  LA R13,ATBLE
  LA R8,TABLE
  LA R12,N6
  LA R1,MASK
  CNOP O,8
  MR R4,R9
  SLDA R4,1
  SRL R5,1
  AR R4,R5
  NR R5,R4
  SLA R4,R1
  ST R0,0(R4,R8)
  XR R5,0(R4,R8)
  XR R0,R5
  XR R5,R0
  XR R0,R5
  BC R4,R11
  B,F15
  LNR R5,R5
  SLR R4,R4
  CLC R5,C1
  BC 11,F25
  SLDL R4,8
  IC R4,0(R4,R13)
  STC R4,PWRD+1
  SRL R5,8
  ALR R5,R10

BASE REGISTER
BRANCH AROUND ID
  SAVE REGISTERS IN HIGH SAVE AREA
  ADDRESS OF HIGH SAVE AREA IN LOW SAVE AR.
  COPY TO R2
  ADDRESS OF LOW SAVE AREA
  ADDRESS OF LOW SAVE AREA IN HIGH SAVE AR.
  LOAD MULTIPLIER, EXPONENT, AND TEST MASK
  CONSTANTS OF THREE ARGUMENTS
  ADDRESSES OF THREE ARGUMENTS
  LOAD STARTING VALUE INTO R5
  NUMBER OF CONSECUTIVE WORDS TO FILL
  CONVERT TO BYTES
  BACKUP ONE WORD IN CALLER'S ARRAY
  INITIAL VALUE FOR INDEX REGISTERS
  ADDRESS OF TABLE OF CONSTANTS
  ADDRESS OF SHUFFLING TABLE
  ADDRESS OF BXLE
  INDEX MASK FOR SHUFFLING
  ALIGN BXLE LOOP FCR SPEED
  FORM PRODUCT OF A AND X(N-1)
  R+ = ADD REMAINDER TO REMAINDER THEREBY
  ADD QUOTIENT DIVISION BY 2**31-1
  PUT X(N) INTO R5
  EXTRACT RIGHT-MOST 7 BITS
  CONVERT TO BYTE OFFSET IN TABLE
  SELECT RANDOM TABLE VALUE
  REPLACE TABLE VALUE WITH X(N)
  EXCHANGE R0 AND R5
  BY THEM WITH EACH OTHER
  SHOULD WE MAKE IT NEGATIVE ?
  POSITIVE, KEEP GCING
  MAKE R5 TRUE NEGATIVE
  CLEAR R4 TO ZERO
  R5 LESS THAN X'68000000' ?
  NO
  SHIFT FIRST 8 BITS OF R5 INTO R4 AS INDEX
  OBTAIN CONSTANT FROM TABLE
  STORE IN SECOND BYTE OF PWRD
  SHIFT REMAINING 24 BITS RIGHT THEN OR ON
  EXPONENT TO MAKE .(24 BITS)/16

```

L6

F15

LLRA3250  
LLRA3260  
LLRA3270  
LLRA3280  
LLRA3290  
LLRA3300  
LLRA3310  
LLRA3320  
LLRA3330  
LLRA3340  
LLRA3350  
LLRA3360  
LLRA3370  
LLRA3380  
LLRA3390  
LLRA3400  
LLRA3410  
LLRA3420  
LLRA3430  
LLRA3440  
LLRA3450  
LLRA3460  
LLRA3470  
LLRA3480  
LLRA3490  
LLRA3500  
LLRA3510  
LLRA3520  
LLRA3530  
LLRA3540  
LLRA3550  
LLRA3560  
LLRA3570  
LLRA3580  
LLRA3590  
LLRA3600  
LLRA3610  
LLRA3620  
LLRA3630  
LLRA3640  
LLRA3650  
LLRA3660  
LLRA3670  
LLRA3680  
LLRA3690  
LLRA3700  
LLRA3710  
LLRA3720  
LLRA3730

\*\*\*\*\* NAVAL POSTGRADUATE SCHOOL RANDOM NUMBER GENERATOR : LLRANDCM \*\*\*\*\*

|     |      |              |                                           |          |
|-----|------|--------------|-------------------------------------------|----------|
| F2S | ST   | R5,0(R7,R6)  | STORE IN CALLER'S ARRAY                   | LLRA3740 |
|     | LE   | FRO,PWRD     | LOAD CHARACTERISTIC TO FLOATING POINT     | LLRA3750 |
|     | AE   | FRO,0(R7,R6) | REGISTER 0 AND ADD FRACTION               | LLRA3760 |
|     | STE  | FRO,0(R7,R6) | STORE NORMAL DEVIATE IN CALLER'S ARRAY    | LLRA3770 |
|     | LR   | R5,R0        | COPY BACK TO R5 FOR NEXT GO AROUND        | LLRA3780 |
|     | CL   | R12          | GO TO BXLE AND CONTINUE                   | LLRA3790 |
|     | BC   | R5,C2        | R5 LESS THAN X'D0000000, ?                | LLRA3800 |
|     | SLDL | 11,F35       | NO                                        | LLRA3810 |
|     | SL   | R4,C1M       | SHIFT FIRST 8 BITS OF R5 INTO R4 AS INDEX | LLRA3820 |
|     | ITC  | R4,0(R4,R13) | SUBTRACT 68                               | LLRA3830 |
|     | SRL  | R4,NWRD+1    | OBTAIN CONSTANT FROM TABLE                | LLRA3840 |
|     | ALR  | R5,8         | STORE IN SECOND BYTE OF NWRD              | LLRA3850 |
|     | ST   | R5,R1C       | SHIFT REMAINING 24 BITS RIGHT THEN OR ON  | LLRA3860 |
|     | LE   | R5,0(R7,R6)  | EXPONENT TO MAKE .(24 BITS)/16            | LLRA3870 |
|     | SEE  | FRO,NWRD     | STORE IN CALLER'S ARRAY                   | LLRA3880 |
|     | STE  | FRO,0(R7,R6) | LOAD CHARACTERISTIC TO FLOATING POINT     | LLRA3890 |
|     | LR   | FRO,0(R7,R6) | REGISTER 0 AND SUBTRACT FRACTION          | LLRA3900 |
|     | CL   | R5,R0        | STORE NORMAL DEVIATE IN CALLER'S ARRAY    | LLRA3910 |
|     | BC   | R12          | COPY BACK TO R5 FOR NEXT GO AROUND        | LLRA3920 |
|     | SLDL | R5,C3        | GO TO BXLE AND CONTINUE                   | LLRA3930 |
|     | SL   | 11,F45       | R5 LESS THAN X'E2F0000, ?                 | LLRA3940 |
|     | ITC  | R4,C2M       | NO                                        | LLRA3950 |
|     | SRL  | R4,NWRD+1    | SHIFT FIRST 12 BITS OF R5 INTO R4         | LLRA3960 |
|     | ALR  | R5,8         | SUBTRACT CE8                              | LLRA3970 |
|     | ST   | R5,0(R7,R6)  | OBTAIN CONSTANT FROM TABLE                | LLRA3980 |
|     | LE   | FRO,PWRD     | STORE IN SECOND BYTE OF PWRD              | LLRA3990 |
|     | AE   | FRO,0(R7,R6) | SHIFT REMAINING 20 BITS RIGHT THEN OR ON  | LLRA4000 |
|     | STE  | R5,R0        | EXPONENT TO MAKE .(20 BITS)/16            | LLRA4010 |
|     | LR   | R12          | STORE IN CALLER'S ARRAY                   | LLRA4020 |
|     | CL   | R5,C4        | LOAD CHARACTERISTIC TO FLOATING POINT     | LLRA4030 |
|     | BC   | 11,F55       | REGISTER 0 AND ADD FRACTION               | LLRA4040 |
|     | SLDL | R4,C3M       | STORE NORMAL DEVIATE IN CALLER'S ARRAY    | LLRA4050 |
|     | SL   | R4,0(R4,R13) | COPY BACK TO R5 FOR NEXT GC AROUND        | LLRA4060 |
|     | ITC  | R4,NWRD+1    | GO TO BXLE AND CONTINUE                   | LLRA4070 |
|     | SRL  | R5,8         | R5 LESS THAN X'F5E0000, ?                 | LLRA4080 |
|     | ALR  | R5,0(R7,R6)  | NO                                        | LLRA4090 |
|     | ST   | FRO,NWRD     | SHIFT FIRST 12 BITS OF R5 INTO R4         | LLRA4100 |
|     | LE   | FRO,0(R7,R6) | SUBTRACT E17                              | LLRA4110 |
|     | SEE  | FRO,0(R7,R6) | OBTAIN CONSTANT FROM TABLE                | LLRA4120 |
|     | STE  | R5,R0        | STORE AS SECOND BYTE OF NWRD              | LLRA4130 |
|     | LR   | R12          | SHIFT REMAINING 20 BITS RIGHT THEN OR ON  | LLRA4140 |
|     | CL   | R5,C5        | EXPONENT TO MAKE .(20 BITS)/16            | LLRA4150 |
|     | BC   | 11,F65       | STORE IN CALLER'S ARRAY                   | LLRA4160 |
|     | SLDL | R4,C4M       | LOAD CHARACTERISTIC TO FLOATING POINT     | LLRA4170 |
|     | SL   | R4,0(R4,R13) | REGISTER 0 AND SUBTRACT FRACTION          | LLRA4180 |
|     | ITC  | R4,NWRD+1    | STORE NORMAL DEVIATE IN CALLER'S ARRAY    | LLRA4190 |
|     | SRL  | R5,8         | COPY BACK TO R5 FOR NEXT GO AROUND        | LLRA4200 |
|     | ALR  | R5,0(R7,R6)  | GO TO BXLE AND CONTINUE                   | LLRA4210 |
|     | ST   | FRO,NWRD     | STORE R5 IN ARGUMENT LIST                 | LLRA4220 |
|     | LE   | FRO,0(R7,R6) |                                           |          |
|     | SEE  | FRO,0(R7,R6) |                                           |          |
|     | STE  | R5,R0        |                                           |          |
|     | LR   | R12          |                                           |          |
|     | CL   | R5,FWRD      |                                           |          |
|     | BC   |              |                                           |          |
|     | SLDL |              |                                           |          |
|     | SL   |              |                                           |          |
|     | ITC  |              |                                           |          |
|     | SRL  |              |                                           |          |
|     | ALR  |              |                                           |          |
|     | ST   |              |                                           |          |
|     | LE   |              |                                           |          |
|     | AE   |              |                                           |          |
|     | STE  |              |                                           |          |
|     | LR   |              |                                           |          |
|     | CL   |              |                                           |          |
|     | BC   |              |                                           |          |
|     | SLDL |              |                                           |          |
|     | SL   |              |                                           |          |
|     | ITC  |              |                                           |          |
|     | SRL  |              |                                           |          |
|     | ALR  |              |                                           |          |
|     | ST   |              |                                           |          |
|     | LE   |              |                                           |          |
|     | AE   |              |                                           |          |
|     | STE  |              |                                           |          |
|     | LR   |              |                                           |          |
|     | CL   |              |                                           |          |
|     | BC   |              |                                           |          |
|     | SLDL |              |                                           |          |
|     | SL   |              |                                           |          |
|     | ITC  |              |                                           |          |
|     | SRL  |              |                                           |          |
|     | ALR  |              |                                           |          |
|     | ST   |              |                                           |          |
|     | LE   |              |                                           |          |
|     | AE   |              |                                           |          |
|     | STE  |              |                                           |          |
|     | LR   |              |                                           |          |
|     | CL   |              |                                           |          |
|     | BC   |              |                                           |          |
|     | SLDL |              |                                           |          |
|     | SL   |              |                                           |          |
|     | ITC  |              |                                           |          |
|     | SRL  |              |                                           |          |
|     | ALR  |              |                                           |          |
|     | ST   |              |                                           |          |
|     | LE   |              |                                           |          |
|     | AE   |              |                                           |          |
|     | STE  |              |                                           |          |
|     | LR   |              |                                           |          |
|     | CL   |              |                                           |          |
|     | BC   |              |                                           |          |
|     | SLDL |              |                                           |          |
|     | SL   |              |                                           |          |
|     | ITC  |              |                                           |          |
|     | SRL  |              |                                           |          |
|     | ALR  |              |                                           |          |
|     | ST   |              |                                           |          |
|     | LE   |              |                                           |          |
|     | AE   |              |                                           |          |
|     | STE  |              |                                           |          |
|     | LR   |              |                                           |          |
|     | CL   |              |                                           |          |
|     | BC   |              |                                           |          |
|     | SLDL |              |                                           |          |
|     | SL   |              |                                           |          |
|     | ITC  |              |                                           |          |
|     | SRL  |              |                                           |          |
|     | ALR  |              |                                           |          |
|     | ST   |              |                                           |          |
|     | LE   |              |                                           |          |
|     | AE   |              |                                           |          |
|     | STE  |              |                                           |          |
|     | LR   |              |                                           |          |
|     | CL   |              |                                           |          |
|     | BC   |              |                                           |          |
|     | SLDL |              |                                           |          |
|     | SL   |              |                                           |          |
|     | ITC  |              |                                           |          |
|     | SRL  |              |                                           |          |
|     | ALR  |              |                                           |          |
|     | ST   |              |                                           |          |
|     | LE   |              |                                           |          |
|     | AE   |              |                                           |          |
|     | STE  |              |                                           |          |
|     | LR   |              |                                           |          |
|     | CL   |              |                                           |          |
|     | BC   |              |                                           |          |
|     | SLDL |              |                                           |          |
|     | SL   |              |                                           |          |
|     | ITC  |              |                                           |          |
|     | SRL  |              |                                           |          |
|     | ALR  |              |                                           |          |
|     | ST   |              |                                           |          |
|     | LE   |              |                                           |          |
|     | AE   |              |                                           |          |
|     | STE  |              |                                           |          |
|     | LR   |              |                                           |          |
|     | CL   |              |                                           |          |
|     | BC   |              |                                           |          |
|     | SLDL |              |                                           |          |
|     | SL   |              |                                           |          |
|     | ITC  |              |                                           |          |
|     | SRL  |              |                                           |          |
|     | ALR  |              |                                           |          |
|     | ST   |              |                                           |          |
|     | LE   |              |                                           |          |
|     | AE   |              |                                           |          |
|     | STE  |              |                                           |          |
|     | LR   |              |                                           |          |
|     | CL   |              |                                           |          |
|     | BC   |              |                                           |          |
|     | SLDL |              |                                           |          |
|     | SL   |              |                                           |          |
|     | ITC  |              |                                           |          |
|     | SRL  |              |                                           |          |
|     | ALR  |              |                                           |          |
|     | ST   |              |                                           |          |
|     | LE   |              |                                           |          |
|     | AE   |              |                                           |          |
|     | STE  |              |                                           |          |
|     | LR   |              |                                           |          |
|     | CL   |              |                                           |          |
|     | BC   |              |                                           |          |
|     | SLDL |              |                                           |          |
|     | SL   |              |                                           |          |
|     | ITC  |              |                                           |          |
|     | SRL  |              |                                           |          |
|     | ALR  |              |                                           |          |
|     | ST   |              |                                           |          |
|     | LE   |              |                                           |          |
|     | AE   |              |                                           |          |
|     | STE  |              |                                           |          |
|     | LR   |              |                                           |          |
|     | CL   |              |                                           |          |
|     | BC   |              |                                           |          |
|     | SLDL |              |                                           |          |
|     | SL   |              |                                           |          |
|     | ITC  |              |                                           |          |
|     | SRL  |              |                                           |          |
|     | ALR  |              |                                           |          |
|     | ST   |              |                                           |          |
|     | LE   |              |                                           |          |
|     | AE   |              |                                           |          |
|     | STE  |              |                                           |          |
|     | LR   |              |                                           |          |
|     | CL   |              |                                           |          |
|     | BC   |              |                                           |          |
|     | SLDL |              |                                           |          |
|     | SL   |              |                                           |          |
|     | ITC  |              |                                           |          |
|     | SRL  |              |                                           |          |
|     | ALR  |              |                                           |          |
|     | ST   |              |                                           |          |
|     | LE   |              |                                           |          |
|     | AE   |              |                                           |          |
|     | STE  |              |                                           |          |
|     | LR   |              |                                           |          |
|     | CL   |              |                                           |          |
|     | BC   |              |                                           |          |
|     | SLDL |              |                                           |          |
|     | SL   |              |                                           |          |
|     | ITC  |              |                                           |          |
|     | SRL  |              |                                           |          |
|     | ALR  |              |                                           |          |
|     | ST   |              |                                           |          |
|     | LE   |              |                                           |          |
|     | AE   |              |                                           |          |
|     | STE  |              |                                           |          |
|     | LR   |              |                                           |          |
|     | CL   |              |                                           |          |
|     | BC   |              |                                           |          |
|     | SLDL |              |                                           |          |
|     | SL   |              |                                           |          |
|     | ITC  |              |                                           |          |
|     | SRL  |              |                                           |          |
|     | ALR  |              |                                           |          |
|     | ST   |              |                                           |          |
|     | LE   |              |                                           |          |
|     | AE   |              |                                           |          |
|     | STE  |              |                                           |          |
|     | LR   |              |                                           |          |
|     | CL   |              |                                           |          |
|     | BC   |              |                                           |          |
|     | SLDL |              |                                           |          |
|     | SL   |              |                                           |          |
|     | ITC  |              |                                           |          |
|     | SRL  |              |                                           |          |
|     | ALR  |              |                                           |          |
|     | ST   |              |                                           |          |
|     | LE   |              |                                           |          |
|     | AE   |              |                                           |          |
|     | STE  |              |                                           |          |
|     | LR   |              |                                           |          |
|     | CL   |              |                                           |          |
|     | BC   |              |                                           |          |
|     | SLDL |              |                                           |          |
|     | SL   |              |                                           |          |
|     | ITC  |              |                                           |          |
|     | SRL  |              |                                           |          |
|     | ALR  |              |                                           |          |
|     | ST   |              |                                           |          |
|     | LE   |              |                                           |          |
|     | AE   |              |                                           |          |
|     | STE  |              |                                           |          |
|     | LR   |              |                                           |          |
|     | CL   |              |                                           |          |
|     | BC   |              |                                           |          |
|     | SLDL |              |                                           |          |
|     | SL   |              |                                           |          |
|     | ITC  |              |                                           |          |
|     | SRL  |              |                                           |          |
|     | ALR  |              |                                           |          |
|     | ST   |              |                                           |          |
|     | LE   |              |                                           |          |
|     | AE   |              |                                           |          |
|     | STE  |              |                                           |          |
|     | LR   |              |                                           |          |
|     | CL   |              |                                           |          |
|     | BC   |              |                                           |          |
|     | SLDL |              |                                           |          |
|     | SL   |              |                                           |          |
|     | ITC  |              |                                           |          |
|     | SRL  |              |                                           |          |
|     | ALR  |              |                                           |          |
|     | ST   |              |                                           |          |
|     | LE   |              |                                           |          |
|     | AE   |              |                                           |          |
|     | STE  |              |                                           |          |
|     | LR   |              |                                           |          |
|     | CL   |              |                                           |          |
|     | BC   |              |                                           |          |
|     | SLDL |              |                                           |          |
|     | SL   |              |                                           |          |
|     | ITC  |              |                                           |          |
|     | SRL  |              |                                           |          |
|     | ALR  |              |                                           |          |
|     | ST   |              |                                           |          |
|     | LE   |              |                                           |          |
|     | AE   |              |                                           |          |
|     | STE  |              |                                           |          |
|     | LR   |              |                                           |          |
|     | CL   |              |                                           |          |
|     | BC   |              |                                           |          |
|     | SLDL |              |                                           |          |
|     | SL   |              |                                           |          |
|     | ITC  |              |                                           |          |
|     | SRL  |              |                                           |          |
|     | ALR  |              |                                           |          |
|     | ST   |              |                                           |          |
|     | LE   |              |                                           |          |
|     | AE   |              |                                           |          |
|     | STE  |              |                                           |          |
|     | LR   |              |                                           |          |
|     | CL   |              |                                           |          |
|     | BC   |              |                                           |          |
|     | SLDL |              |                                           |          |
|     | SL   |              |                                           |          |
|     | ITC  |              |                                           |          |
|     | SRL  |              |                                           |          |
|     | ALR  |              |                                           |          |
|     | ST   |              |                                           |          |
|     | LE   |              |                                           |          |
|     | AE   |              |                                           |          |
|     | STE  |              |                                           |          |
|     | LR   |              |                                           |          |
|     | CL   |              |                                           |          |
|     | BC   |              |                                           |          |
|     | SLDL |              |                                           |          |
|     | SL   |              |                                           |          |
|     | ITC  |              |                                           |          |
|     | SRL  |              |                                           |          |
|     | ALR  |              |                                           |          |
|     | ST   |              |                                           |          |
|     | LE   |              |                                           |          |

\*\*\*\*\* NAVAL POSTGRADUATE SCHOOL RANDOM NUMBER GENERATOR : LLRANDOM \*\*\*\*\*

```

ST      RQ,XWRD
LA      R13,SA2
LA      R1,FLIST
LA      R8,R15,NOR
LA      R15,R15
BALR    R14,R15
LSTE    R15,R9
L        R10,0(R7,R6)
LA      R5,XWRD
LA      R13,ATBLE
LA      R8,TABLE
LA      R1,MASK
BXLE    R7,R2,L6
L        R13,SA2+4
L        R1,24(R13)
L        R4,0(R1)
LSTE    R5,0(R4)
L        R14,R12,12(R13)
BCR     15,R14

PASS STARTING VALUE
LOAD LOW SAVE AREA POINTER
ARGUMENT LIST FOR CALL TC RNORTH
COPY BASE FOR BALR LINKAGE
ADDRESS OF FUNCTION SUBROUTINE RNORTH
BRANCH TO RNORTH
RESTORE BASE REGISTER
STORE NORMAL VALUE
NEW STARTING VALUE
RESTORE R13 TO TABLE OF CONSTANTS
RESTORE R8 TO ADDRESS OF SHUFFLING TABLE
RESTORE R1 TO INDEX MASK
LOCP AROUND AGAIN
RESTORE HIGH SAVE AREA PCINTER
GET ARGUMENT LIST POINTER AGAIN
GET STARTING VALUE ADDRESS AGAIN
STORE AS STARTING VALUE FOR NEXT CALL
RESTORE THE REGISTERS
RETURN

```

LLRA4230  
 LLRA4240  
 LLRA4250  
 LLRA4260  
 LLRA4270  
 LLRA4280  
 LLRA4290  
 LLRA4300  
 LLRA4310  
 LLRA4320  
 LLRA4330  
 LLRA4340  
 LLRA4350  
 LLRA4360  
 LLRA4370  
 LLRA4380  
 LLRA4390  
 LLRA4400  
 LLRA4410

N6

\*\*\*\*\* NAVAL POSTGRADUATE SCHOOL RANDOM NUMBER GENERATOR : LLRANDOM \*\*\*\*\*

```

*      ENTRY POINT :  EXPON
*
*      EXPON
*      CNOP 0,8
*      USING EXPON,R15
*      B 10(,R15)
*      DC AL1(5)
*      DC CL5,EXPON*
*      ST R14,R12,12(R13)
*      LR R13,SA2+4
*      LA R13,SA2
*      ST R13,8(,R2)
*      LM R9,R11,A75N
*      LA R2,4
*      LM R5,R7,0(R1)
*      L R5,0(,R5)
*      L R3,0(,R7)
*      L R3,2
*      L R6,R2
*      L R7,R2
*      LA R13,BTBL
*      LA R12,N7
*      CNOP 0,8
*      MR R4,R9
*      SLDA R4,1
*      SRL R5,1
*      AR R4,R5
*      LR R0,R5
*      LR R4,R11
*      BC R5,E1
*      LNR R5,R4
*      CCL R4,R4
*      BC R5,D1
*      SLDL R4,8
*      IC R4,0(R4,R13)
*      STC R4,PWRD+1
*      SRL R5,8
*      ALR R5,R10
*      ST R5,0(R7,R6)
*      LE FRO,PWRD
*      AE FRO,0(R7,R6)
*      STE FRO,C(R7,R6)
*      LBR R5,R0
*      CCL R12
*      R5,D2
*      L1,E3

```

BASE REGISTER  
BRANCH AROUND ID

SAVE REGISTERS IN HIGH SAVE AREA  
COPY TO R2  
ADDRESS OF LOW SAVE AREA  
ADDRESS OF LOW SAVE AREA, IN HIGH SAVE AR.  
LCAD MULTIPLIER, EXPONENT, AND TEST MASK  
CONSTANTS OF THREE ARGUMENTS  
ADDRESSES OF THREE ARGUMENTS  
LOAD STARTING VALUE INTO R5  
NUMBER OF CONSECUTIVE WORDS TO FILL  
CONVERT TO BYTES IN CALLER'S ARRAY  
BACKUP ONE WORD FOR INDEX REGISTER  
INITIAL VALUE OF INDEX REGISTER  
ADDRESS OF TABLE OF CONSTANTS  
ALIGN BXLE LOOP FOR SPEED  
FORM PRODUCT OF A AND X(N-1)  
R4 = REMAINDER ; R5 = QUOTIENT  
ADD QUOTIENT TO REMAINDER THEREBY  
SIMULATING DIVISION BY 2\*\*31-1  
PUT X(N) INTO R5  
COPY R5 INTO R0 FOR NOW  
SHOULD WE MAKE IT NEGATIVE ?  
POSITIVE, KEEP GOING  
MAKE R5 TRUE NEGATIVE  
CLEAR R4 TO ZERO  
R5 LESS THAN X.D5000000 ?  
NO  
SHIFT FIRST 8 BITS OF R5 INTO R4 AS INDEX  
OBTAIN CONSTANT FROM TABLE  
STORE IN SECOND BYTE OF PWRD  
SHIFT REMAINING 24 BITS RIGHT THEN OR ON  
EXPONENT TO MAKE .(24 BITS)/16  
STORE IN CALLER'S ARRAY  
LCAD CHARACTERISTIC TO FLOATING POINT  
REGISTER 0 AND ADD FRACTION  
STORE EXPONENTIAL DEVIATE IN ARRAY  
COPY BACK TO R5 FOR NEXT GO AROUND  
GO TO BXLE AND CCNTINUE  
R5 LESS THAN X.F17000000 ?  
NO

L7

E1

E2

LLRA4430  
LLRA4440  
LLRA4450  
LLRA4460  
LLRA4470  
LLRA4480  
LLRA4490  
LLRA4500  
LLRA4510  
LLRA4520  
LLRA4530  
LLRA4540  
LLRA4550  
LLRA4560  
LLRA4570  
LLRA4580  
LLRA4590  
LLRA4600  
LLRA4610  
LLRA4620  
LLRA4630  
LLRA4640  
LLRA4650  
LLRA4660  
LLRA4670  
LLRA4680  
LLRA4690  
LLRA4700  
LLRA4710  
LLRA4720  
LLRA4730  
LLRA4740  
LLRA4750  
LLRA4760  
LLRA4770  
LLRA4780  
LLRA4790  
LLRA4800  
LLRA4810  
LLRA4820  
LLRA4830  
LLRA4840  
LLRA4850  
LLRA4860  
LLRA4870  
LLRA4880  
LLRA4890  
LLRA4900  
LLRA4910



\*\*\*\*\* NAVAL POSTGRADUATE SCHOOL RANDOM NUMBER GENERATOR : LLRANDOM \*\*\*\*\*

```

* * *
* ENTRY POINT : SEXPON
*
SEXPON      CNOP 0,8
            USING SEXPON,R15
            B 12(,R15)
            DC AL1(6)
            DC CL6,SEXPON'
            STM R14,R12,12(R13)
            ST  R13,SA2+4
            LR  R2,R13
            LA  R13,SA2
            ST  R13,8(,R2)
            LM  R9,R11,A75N
            LA  R2,4
            LA  R5,R7,0(R1)
            LA  R5,0(,R5)
            LA  R3,0(,R7)
            LA  R3,2
            SLA R6,R2
            SR  R7,R2
            LR  R13,BTBL
            LA  R8,BTBL
            LA  R12,N8
            LA  R1,MASK
            LCNOP 0,8
            MRDA R4,R9
            SRL R4,1
            AR  R5,15
            LR  R5,R4
            LR  R4,R1
            SLA R4,2
            LT  R0,0(R4,R8)
            ST  R5,0(R4,R8)
            XR  R0,R5
            XR  R5,R0
            XR  R0,R5
            XR  R4,R11
            BC  8,E1S
            LNR R5,R5
            SLR R4,R4
            CL  R5,D1
            BC  11,E2S
            SLDL R4,8
            IC  R4,0(R4,R13)
            STC R4,PWRD+1
            SRL R5,8
            ALR R5,R10
            E1S

L8
            BASE REGISTER
            BRANCH AROUND ID
            SAVE REGISTERS IN HIGH SAVE AREA
            ADDRESS OF HIGH SAVE AREA IN LOW SAVE AR.
            COPY TO R2
            ADDRESS OF LOW SAVE AREA
            ADDRESS OF LOW SAVE AREA IN HIGH SAVE AR.
            LOAD MULTIPLIER, EXPONENT, AND TEST MASK
            CONSTANT FOR BXLE
            ADDRESSES OF THREE ARGUMENTS
            LOAD STARTING VALUE IN R5
            NUMBER OF CONSECUTIVE WORDS TO FILL
            CONVERT TO BYTES
            BACKUP ONE WORD IN CALLER'S ARRAY
            INITIAL VALUE FOR INDEX REGISTER
            ADDRESS OF TABLE OF CONSTANTS
            ADDRESS OF SHUFFLING TABLE
            ADDRESS OF BXLE SHUFFLING
            ALIGN BXLE LOOP FOR SPEED
            FORM PRODUCT OF A AND X(N-1)
            R4 = QUOTIENT ; R5 = REMAINDER THEREBY
            ADD QUOTIENT TO REMAINDER BY 2**31-1
            PUT X(N) INTO R5
            EXTRACT RIGHT-MOST 7 BITS
            CONVERT TO BYTE OFFSET IN TABLE
            SELECT RANDOM TABLE VALUE
            REPLACE TABLE VALUE WITH X(N)
            EXCHANGE R0 AND R5
            BY EXCLUSIVE OR'ING
            SHOULD WE MAKE IT NEGATIVE ?
            POSITIVE, KEEP GOING
            MAKE R5 TRUE NEGATIVE
            CLEAR R4 TO ZERO
            R5 LESS THAN X'D5000000' ?
            NO
            SHIFT FIRST 8 BITS OF R5 INTO R4 AS INDEX
            OBTAIN CONSTANT FROM TABLE
            STORE IN SECOND BYTE OF PWRD
            SHIFT REMAINING 24 BITS RIGHT THEN OR ON
            EXPONENT TO MAKE .(24 BITS)/16

```

LLRA5230  
 LLRA5240  
 LLRA5250  
 LLRA5260  
 LLRA5270  
 LLRA5280  
 LLRA5290  
 LLRA5300  
 LLRA5310  
 LLRA5320  
 LLRA5330  
 LLRA5340  
 LLRA5350  
 LLRA5360  
 LLRA5370  
 LLRA5380  
 LLRA5390  
 LLRA5400  
 LLRA5410  
 LLRA5420  
 LLRA5430  
 LLRA5440  
 LLRA5450  
 LLRA5460  
 LLRA5470  
 LLRA5480  
 LLRA5490  
 LLRA5500  
 LLRA5510  
 LLRA5520  
 LLRA5530  
 LLRA5540  
 LLRA5550  
 LLRA5560  
 LLRA5570  
 LLRA5580  
 LLRA5590  
 LLRA5600  
 LLRA5610  
 LLRA5620  
 LLRA5630  
 LLRA5640  
 LLRA5650  
 LLRA5660  
 LLRA5670  
 LLRA5680  
 LLRA5690  
 LLRA5700  
 LLRA5710

\*\*\*\*\* NAVAL POSTGRADUATE SCHOOL RANDOM NUMBER GENERATOR : LLRANDCM \*\*\*\*\*

```

E2S
ST  R5,0(R7,R6)
LE  FRO,PWRD
AE  FRO,0(R7,R6)
ST  FRO,0(R7,R6)
LR  R5,RO
BR  R12
BC  R5,D2
SLDL 11,E3S
SL  R4,12
IC  R4,DIM
STC  R4,0(R4,R13)
SKL  R4,PWRD+1
ALR  R5,8
ST  R5,R10
LE  R5,0(R7,R6)
AE  FRO,PWRD
ST  FRO,0(R7,R6)
LR  R5,RO
BR  R12
ST  R5,EWRD
LA  R0,XWRD
LA  R1,SA2
LA  R1,ELIST
LA  R8,R15
L  R15,AREXP
BALR R14,R15
ST  R15,R8
LA  FRO,0(R7,R6)
LA  R5,XWRD
LA  R13,BTABLE
LA  R8,BTABLE
LA  R1,MASK
BXLE R7,R2,L8
L  R13,SA2+4
L  R1,24(R13)
L  R4,0(R1)
L  R5,0(R4)
LM  R14,R12,12(R13)
BCR 15,R14

E3S
STORE IN CALLER'S ARRAY
LOAD CHARACTERISTIC TO FLOATING POINT
REGISTER 0 AND ADD FRACTCN
STORE EXPONENTIAL DEVIATE IN ARRAY
COPY BACK TO R5 FOR NEXT GC AROUND
GO TO BXLE AND CCNTINUE
R5 LESS THAN X.F1700000, ?
NO
SHIFT FIRST 12 BITS OF R5 INTO R4
SUBTRACT CFF
OBTAIN CONSTANT FROM TABLE
STORE AS SECOND BYTE OF PWRD
SHIFT REMAINING 20 BITS RIGHT THEN OR ON
EXPONENT TO MAKE .(20 BITS)/16
STORE IN CALLER'S ARRAY
LOAD CHARACTERISTIC TO FLOATING POINT
REGISTER 0 AND ADD FRACTCN
STORE EXPONENTIAL DEVIATE IN ARRAY
COPY BACK TO R5 FOR NEXT GC AROUND
GO TO BXLE AND CCNTINUE
STORE R5 IN ARGUMENT LIST
PASS STARTING VALUE
LOAD LOW SAVE AREA POINTER
ARGUMENT LIST FOR CALL TO REXPTH
COPY BASE FOR BALR LINKAGE
ADDRESS OF FUNCTION SUBROUTINE REXPTH
BRANCH TO REXPTH
STORE EXPONENTIAL DEVIATE IN ARRAY
NEW STORE STARTING VALUE
RESTORE R13 TO TABLE OF CCNTENTS
RESTORE R8 TO ADDRESS OF SHUFFLING TABLE
RESTORE R1 TO INDEX MASK
LOOP AROUND AGAIN
RESTORE HIGH SAVE AREA PCINTER
GET ARGUMENT LIST POINTER AGAIN
GET STARTING VALUE ADDRESS AGAIN
STORE AS STARTING VALUE FOR NEXT CALL
RESTORE THE REGISTERS
RETURN

LLRA5720
LLRA5730
LLRA5740
LLRA5750
LLRA5760
LLRA5770
LLRA5780
LLRA5790
LLRA5800
LLRA5810
LLRA5820
LLRA5830
LLRA5840
LLRA5850
LLRA5860
LLRA5870
LLRA5880
LLRA5890
LLRA5900
LLRA5910
LLRA5920
LLRA5930
LLRA5940
LLRA5950
LLRA5960
LLRA5970
LLRA5980
LLRA5990
LLRA6000
LLRA6010
LLRA6020
LLRA6030
LLRA6040
LLRA6050
LLRA6060
LLRA6070
LLRA6080
LLRA6090
LLRA6100
LLRA6110

```

N8



\*\*\*\*\* NAVAL POSTGRADUATE SCHOOL RANDOM NUMBER GENERATOR : LLRANDOM \*\*\*\*\*

|    |             |             |              |              |          |
|----|-------------|-------------|--------------|--------------|----------|
| DC | X'76839B19' | X'0DB1782F' | X'69185168'  | X'788F0F60'  | LLRA6620 |
| DC | X'42A0D52'  | X'20AB7919' | X'2070A25E'  | X'7611BBA9'  | LLRA6630 |
| DC | X'474849DF' | X'7C880DEB' | X'53D023CE'  | X'6D9FEFF46' | LLRA6640 |
| DC | X'7D91C530' | X'0B4788B4' | X'5C26E9F8'  | X'4D8F5AC7'  | LLRA6650 |
| DC | X'7366FDCB' | X'305629D9' | X'2A1C7AF8'  | X'1C6DCB3R'  | LLRA6660 |
| DC | X'4DE86B25' | X'312260FF' | X'58A65DC3C' | X'5EDDBA81E' | LLRA6670 |
| DC | X'5066686D' | X'09440578' | X'OFD6671D'  | X'555C53EE2' | LLRA6680 |
| DC | X'2ED81E2F' | X'44AB0388' | X'66EEFE57'  | X'52C5A3F8'  | LLRA6690 |
| DC | X'6D8DDF2B' | X'685F1A04' | X'061B4869'  | X'0CAFCD12A' | LLRA6700 |
| DC | X'3E60285D' | X'3A3282F8' | X'681DD5ED'  | X'1B891EB4'  | LLRA6710 |
| DC | X'0F752CD0' | X'1D54304C' | X'4483AA7D'  | X'4OD52733'  | LLRA6720 |
| DC | X'148D6223' | X'58142A4B' | X'303AAF44'  | X'6949EC28'  | LLRA6730 |
| DC | X'6494F236' | X'403AC1E7' | X'719CF6DA'  | X'13983EF75' | LLRA6740 |
| DC | X'52464E2F' | X'607A98FD' | X'76D4AB55'  | X'3ED2E3CF'  | LLRA6750 |
| DC | X'67625F57' | X'360BA8E2' | X'4D8A3824'  | X'0ACD66BD'  | LLRA6760 |
| DC | X'465A79D1' | X'22648308' | X'249A5E43'  | X'412497D9'  | LLRA6770 |
| DC | X'3A081DB4' | X'34AF002B' | X'53D28497'  | X'26A0D19D'  | LLRA6780 |
| DC | X'521C9024' | X'6DFAC836' | X'5D1F4E1C'  | X'65AA201A'  | LLRA6790 |
| DC | X'2213B94F' | X'0D0955C'  | X'0D25924E'  | X'2DD327FE'  | LLRA6800 |
| DC | X'5DD061F1' | X'1AD48B0A' | X'0CB02L6F'  | X'6CC25FB7'  | LLRA6810 |
| DC | X'02D9E6D8' | X'694B7943' | X'5E31F4A9'  | X'7AF65CAE'  | LLRA6820 |
| DC | X'00010202' | X'03030303' | X'04040404'  | X'0AE725FE'  | LLRA6830 |
| DC | X'0AGA0A0E' | X'0E0E1217' | X'00000000'  | X'04090A0A'  | LLRA6840 |
| DC | X'01010202' | X'02020303' | X'04050505'  | X'0C1C101'   | LLRA6850 |
| DC | X'0606C607' | X'07070707' | X'08080808'  | X'05C5C606'  | LLRA6860 |
| DC | X'0908080B' | X'0B0C0C0C' | X'0C0D0D0D'  | X'0DCEC909'  | LLRA6870 |
| DC | X'0F101010' | X'11111112' | X'12131314'  | X'0DCEC90F'  | LLRA6880 |
| DC | X'16171819' | X'1A1B1C1D' | X'15050505'  | X'14151516'  | LLRA6890 |
| DC | X'05050606' | X'06060606' | X'06070707'  | X'05C50505'  | LLRA6900 |
| DC | X'0B0B0B0B' | X'0B0B0B0B' | X'0B0C0C0C'  | X'07C7C808'  | LLRA6910 |
| DC | X'0F0F0F0F' | X'0F0F0F0F' | X'0F101010'  | X'0C0C0D0D'  | LLRA6920 |
| DC | X'11111113' | X'13131313' | X'13131313'  | X'1C1C1314'  | LLRA6930 |
| DC | X'14141414' | X'14141414' | X'15151515'  | X'13131314'  | LLRA6940 |
| DC | X'18181818' | X'18181818' | X'18181818'  | X'15161618'  | LLRA6950 |
| DC | X'19191919' | X'19191A1A' | X'1A1A1A1A'  | X'15151919'  | LLRA6960 |
| DC | X'1B1B1C1C' | X'1E1E1E1E' | X'1A1A1A1A'  | X'1A1E1E1E'  | LLRA6970 |
| DC | X'1E1E1E1E' | X'1F1F1F1F' | X'1E1E1E1E'  | X'1E1E1E1E'  | LLRA6980 |
| DC | X'20202020' | X'20202020' | X'20202020'  | X'1F1F1F1F'  | LLRA6990 |
| DC | X'21212121' | X'21212222' | X'22222222'  | X'21212121'  | LLRA7000 |
| DC | X'23232323' | X'23232324' | X'24242424'  | X'22222223'  | LLRA7010 |
| DC | X'25252525' | X'26262626' | X'26272727'  | X'24242525'  | LLRA7020 |
| DC | X'29292929' | X'2A2A2A2A' | X'26272727'  | X'27282828'  | LLRA7030 |
| DC | X'00000000' | X'00000000' | X'00000000'  | X'000000C1'  | LLRA7040 |
| DC | X'0101C101' | X'01010101' | X'01010101'  | X'02020202'  | LLRA7050 |
| DC | X'02020202' | X'02030303' | X'03030606'  | X'06C6C608'  | LLRA7060 |
| DC | X'08080808' | X'0808080A' | X'0A0A0A0A'  | X'0A0A0A0C'  | LLRA7070 |
| DC | X'0C0C0C0C' | X'0C0E0E11' | X'11515151'  | X'15152020'  | LLRA7080 |
| DC | X'2B010202' | X'02020303' | X'03030303'  | X'03C4C404'  | LLRA7100 |

A T B L E

B T B L E

