1-1-1980

A study of random number generators - RANF and URAND.

Kanchan Kumar

A STUDY OF RANDOM NUMBER GENERATORS
- RANF AND URAND

by

Kanchan Kumar

A Thesis

Presented to the Graduate Committee

of Lehigh University

in Candidacy for the Degree of

Master of Science

in

Computer Science

May

1980

CERTIFICATE OF APPROVAL

This thesis is accepted and approved in partial fulfillment of the requirements for the degree of Master of Science.

May 7, 1980
(date)

John W. Adams
Professor in Charge

Samuel L. Gulden
Head of the Division

ACKNOWLEDGEMENTS

I would like to express sincere thanks to my supervisor, Professor John W. Adams for suggesting the topic of this report, and his inspiring guidance throughout the course of this work.

Help from department of Mathematics and department of Industrial Engineering is also acknowledged.

Also, I would like to thank Judy Arroyo for an excellent typing job.

TABLE OF CONTENTS

LIST OF TABLES

ABSTRACT

Randomness of the uniform random numbers generated by CDC 6400 generator - RANF and by a subroutine URAND is studied and tested. The desired statistical properties of the random numbers generated by these two generators are studied using five empirical tests (frequency test, serial test, poker test, gap test and run test). A chi-square (χ^2) statistic is calculated to ascertain the randomness of uniform random numbers generated by RANF and URAND.

Random numbers generated by both the generators have passed all the five tests satisfactorily. However, the chi-square statistics show that RANF is slightly better than URAND.

CHAPTER I - INTRODUCTION

What is a random number? In a rigorous sense, there is no such thing as a random number. Rather, there is a sequence of independent random numbers with a specified distribution. This means loosely that each number was obtained merely by chance having nothing to do with other numbers of the sequence, and that each number has a specified probability of falling in any given range.

Sequences generated in a deterministic way are usually called pseudo-random or quasi-random sequences in the highbrow technical literature, but it can be used as a random sequence for all practical applications.

1.1 APPLICATIONS

Numbers which are "chosen at random" are useful in a very wide variety of applications. For example:

1. Simulation - When a computer is used to simulate natural phenomena, random numbers are required to simulate the uncertainties of the new world. For example, particles are subject to random collisions, people come into a bank or store at random intervals etc. Many other problems in business, industry or science

which involve variables that can only be described statistically can be solved only by employing random numbers.

2. Numerical analysis - Many techniques use the random numbers to solve complicated numerical problems. For example, evaluation of definite integrals, the solution of ordinary and partial differential equations, matrix inversion, problems involving permutations and combinations of variables and certainly many standard statistics problems.

3. Computer programming - Random numbers make a good source of data for testing the effectiveness of computer algorithms.

4. Sampling - Often it is impractical to examine all possible cases, but a random sample will provide insight into most of the questions.

5. Decision making - Sometimes, it is important to make a completely "unbiased" decision, for that you employ random numbers. Randomness is also an essential part of optimal strategies in the theory of games

These are the few applications of the random numbers. "Monte Carlo method" is a general term used to describe any algorithm that employs random numbers.

1.2 HISTORY OF RANDOM NUMBERS AND LITERATURE SURVEY

The generating of random numbers by some physical phenomenon is not a recent development. The technique supposedly began in 1777 with the famous Buffon needle experiment, used to calculate an estimate of π. Development of random number generation techniques is considered to have originated in 1908 with the connection to sampling estimates leading to the t-distribution as a small sample method.

At first, people who needed random numbers in their scientific work would draw balls out of a well-stirred urn or would roll dice or deal out cards. In 1927, a table of over 40,000 random digits, taken at random from census reports, was published by L. H. C. Tippett. Since then, a number of special machines for mechanically generating random numbers have been built. The first such machine was used by M. G. Kendall and B. Babington-Smith [4,5] to produce a table of 100,000 random digits in 1939, and in 1955 the RAND Corporation [11] published its well known table of a million random digits obtained with the help of another special machine. One distinguishing feature of this digit table is its size. Before this, for numerous problems, the largest existing table

Kendall and Smith would have had to be used many times over, with the consequent dangers of introducing unwanted correlations. A famous random-number machine, called ERNIE, picks the winning numbers in the British Premium Saving Bonds lottery [4,5].

The development of the digital computer has led to an increasing need for the representation of phenomena which can be assumed to be random i.e. the resulting outcomes are equally likely and independent. Many computer algorithms has been developed and tested to generate random numbers.

Marsaglia and Bray [8] have discussed some on-line random number generators i.e. generators requiring a single FORTRAN instruction. The advantages are speed (avoiding linkage to and from subroutine), convenience and versatility.

In 1973, Landis and Feinstein [7] studied the random number generated by computer and from the RAND tables. Tests are described and they concluded that computer generated numbers are just as random as those available in the RAND tables that are accepted as a standard source of random numbers. Also Yven [12] in 1977 has done the testing of random number generators by Walsh transformation.

In 1979 Schrage [10] has described a more portable FORTRAN implementation of a random number generator which produces a sequence of random integers that is machine independent as long as the machine can represent all integers in the interval [-2**31+1, 2**31-1].

There are many other papers and references. Nance and Overstreet [9] have provided a very good complete bibliography on random number generations in 1972. The text by Donald E. Knuth [6] provides the most complete and technically sound treatment of random number generation.

CHAPTER II - GENERATION OF RANDOM NUMBERS

2.1 INTRODUCTION

Strictly speaking, a random number exists only as the result of a random process. For example, a mechanical or electronic device, a perfect roulette wheel, supply truly random numbers on demand. But this is not done because (a) nature tends to be systematic, so that construction and maintenance of such a device which must output millions of times in an unsystematic manner, and at a millisecond or even microsecond speed, is not at all cheap or easy to build; (b) it is some times desirable to repeat a calculation exactly which is not possible if a random device is involved.

Another alternative would be to store random numbers, say from RAND tables [11] on some storage device and read in as required. But this is not feasible because it is too slow and due to the restriction of memory space and input time requirement. Also, one might object, reading the same random numbers for every problem.

The inadequacy of these methods led to an interest in the production of random numbers using the arithmetic operations of a computer. By this method random numbers are available when they are needed and are perfectly reproducible for program checkout. Since such numbers are generated by an algorithm, they are not truly random, but for all practical purposes they can be used as random numbers. John vonNeumann first suggested this approach in about 1946 using the middle-square method, but the sequence tends to get into a short cycle of repeating elements. There is another algorithm called super-random number generator, which changes a 10-digit decimal number to the number which should come next in a supposedly random sequence, but it is also not very good at all. There are many other algorithms.

Any sequence of numbers produced by a subroutine with finite input will eventually repeat, since the computer has only a finite number of stable states. To get a long period for the repeating sequence, the two principal number theory ideas, congruences and power residues are employed.

2.2 UNIFORM RANDOM NUMBERS

Those random numbers which are uniformly distributed between zero and one. An almost universally used method of generating uniform random numbers is to select a function f that maps the numbers into themselves.

The commonest function f has the form

$$f(x_n) = ax_n + c \pmod{m} \quad n \geq 0$$

where, $x_0 \geq 0$ some starting value, $a \geq 0$, $c \geq 0$, $m > x_0$, $m > a$, $m > c$.

This gives a sequence called a linear congruential sequence. But congruential sequences always get into a loop, i.e. there is ultimately a cycle of numbers which is repeated endlessly. But, by the appropriate choice of x_0, a, c and m, one can ensure a fairly long period. Knuth [6] summarizes the number theory needed to pick appropriate values of x_0, a, c and m.

1. x_0 can be arbitrary. Different values of x_0 yield different sequences. If the program is run several times and a different source of random numbers is desired each time, set x_0 to the last value attained by x on the preceding run; or set x_0 to the current date and time.

2. The number m should be large. It may be taken as the computer's word size.

3. Pick a to have three properties:

(i) If m is a power of 2 (binary machine) pick a so that a mod 8 = 5. If m is a power of 10 (decimal machine) pick a so that a mod 200 = 21.

(ii) $m/100 < a < m - \sqrt{m}$

(iii) the digits in the binary or decimal representation of a should not have a simple, regular pattern.

4. Choose c an odd number and such that

$$c/m \approx \frac{1}{2} - \frac{1}{6}\sqrt{3} \approx 0.21132 \ .$$

The least significant digits of x_k are not very random, so decisions based on the number x_k should always be primarily influenced by the most significant digits. For example, if you want to choose at random 1 of 16 possible outcomes by use of x_k, use the most significant bits of x_k.

OTHER METHODS

1. The linear congruential method generalized to a quadratic congruential method.

$$f(x_n) = (dx_n^2 + ax_n + c) \bmod m \ .$$

2. The Fibonacci series method.

$$f(x_n) = (x_n + x_{n-1}) \bmod m$$

This method is fast, easy and the period is satisfactorily long but results are not good statistically.

3. Green, Smith and Klem [2] in 1959 have given the following function for generating random numbers.

$$f(x_n) = (x_n + x_{n-k}) \bmod m$$

where k is comparatively large value.

Numerous other methods have been devised. Most are slower and more cumbersome.

2.3 OTHER TYPES OF RANDOM NUMBERS

Often one needs numbers that have a specified statistical distribution other than a uniform distribution on the unit interval. For example, if one wants to make a random choice between k alternatives one will need a random integer between 1 and k. On the other hand, a random number with the exponential distribution is desired in a simulation process which calls for a random waiting time between occurrences of independent events.

It so happens that these more general problems are easily solved once a supply of uniformly distributed random numbers is obtained because in principle, any of these other random quantities may be obtained from the uniform deviates. There are number of important random tricks which may be used to perform these manipulations efficiently on a computer. Knuth [6] has summarized the best known techniques. A study of these techniques also gives some insight into the proper use of random numbers in any Monto Carlo application.

CHAPTER III - TEST FOR RANDOM NUMBERS

The main purpose is to obtain a sequence of random numbers which behaves as if they are random. By making the period of a sequence so long that for practical purposes it never will repeat, is an important criterion, but it by no means guarantees that the sequence will be useful in applications. The theory of statistics provides some quantitative measures for randomness.

3.1 INTRODUCTION

If a sequence behaves randomly with respect to tests $T_1, T_2, \ldots, T_n$, one cannot be sure in general that it will not be a miserable failure when it is subjected to a further test T_{n+1}; yet each test gives more and more confidence in the randomness of a sequence. In empirical tests, the computer manipulates the groups of numbers of the sequence and evaluates a certain statistic. The Chi-square test (^{2} test) is the best known test and it is the basic method which is used in connection with many other tests. Knuth [6] has described and discussed many theoretical tests by using number-theoretical methods based on the recurrence rule.

Chi-Square Test - In general, suppose that every observation can fall into on of k categories. Take n independent observations (where n is large). Let p_s be the probability that each observation falls into category s, and let y_s be the number of observations which actually do fall into category s, then the statistic

$$V = \sum_{1 \leq s \leq k} \frac{(y_s - np_s)^2}{np_s}$$

is called "Chi-Square" statistic with $k-1$ degrees of freedom.

Using the fact that $\sum_{1 \leq s \leq k} y_s = n$ and $\sum_{1 \leq s \leq k} p_s = 1$, one gets

$$V = \frac{1}{n} \sum_{1 \leq s \leq k} \left(\frac{y_s^2}{p_s}\right) - n$$

which is more easy to use.

3.2 EMPIRICAL TESTS

There are many different kinds of tests that have been applied to random digits in order to investigate their randomness. In this section five kinds of specific tests, which I have used in my study, are described. Each test is applied to N digits.

1. <u>FREQUENCY TEST</u> - All the digits 0-9 should occur an approximately equal number of times in a sequence of random digits. In this test, for each digit d, $0 \leq d \leq 9$, count the number of times it occurred and then apply the chi-square test with 9 d.f.

2. <u>SERIAL TEST</u> - More generally, pairs of successive digits should be uniformly distributed in an independent manner. To make this test, count the number of times pair (q,r) occurred for each possible pair (q,r) with $0 \leq q, r \leq q$ and then apply chi-square test with 90 d.f.

3. <u>POKER TEST</u> - In this test, digits are arranged in blocks of five and count the number of times the following seven patterns each occurred.

- i) All different: abcde
- ii) One pair: aabcd
- iii) Two pairs: aabbc
- iv) Three of a kind: aaabc
- v) Full house: aaabb
- vi) Four of a kind: aaaab
- vii) Five of a kind: aaaaa

Then a chi-square test with 6 d.f. is applied.

4. <u>GAP TEST</u> - This test is used to examine the length of "gaps" between the same digits in the series of random numbers. For instance, if one takes a digit, say zero, in about one-tenth of the cases the first zero will be followed immediately by a second zero, and there will be no gap; in about nine-hundredth of the cases there will be one digit between two zeros and give gap of length one and so on.

If α and β are two real numbers such that $0 \leq \alpha < \beta \leq 1$, consider the length of consecutive sub-series $d_j, d_{j+1}, \ldots, d_{j+r}$ in which d_{j+r} lies between α and β but the other d's do not; this subseries of $r+1$ digits represents a gap of length r.

Suppose one counts gaps of length $0,1,\ldots,t-1$ and the number of gaps of length $\geq t$ then the probabilities can be calculated as follows.

$$p_r = p(1-p)^r \quad \text{for} \quad r = 0,1,\ldots,t-1$$

$$p_t = (1-p)^t$$

where $p = \beta-\alpha$, the probability that $\alpha \leq d_j \leq \beta$. A chi-square test is then applied with t d.f.

5. <u>RUN TEST</u> - In this, one examines the length of monotone subsequences of the original sequence, i.e. segments which are increasing or decreasing.

For N numbers $u_1, \ldots, u_N$, write an N-1 bit binary sequence whose n'th term is zero if $u_n < u_{n+1}$ and is 1 if $u_n > u_{n+1}$. A sequence of k zeroes, bracketed by ones at each end, forms a run of zeroes of length k; similarly for runs of ones. The test involves counting the number of occurrences of runs of different lengths and comparing to expected results. For N numbers, the expected frequency can be calculated as follows [3].

$$2\left\{(k^2+3k+1)N-(k^3+3k^2-k-4)\right\}/(k+3)!$$

gives runs of length k for $k < N-1$ and $2/N!$ runs of length N-1. $(2N-1)/3$ gives the total number of runs.

This test is done on successive numbers rather than on digits where as all other four tests described above have been applied to the digits of random numbers.

CHAPTER IV - RANDOM NUMBER GENERATOR - RANF AND URAND

In my study, I have used two random number generators.

1. The CDC 6400 random number generator, RANF

2. URAND in ANSI standard FORTRAN programmed by Forsythe, Malcolm and Moler [1]. This subroutine is designed to be relatively machine independent.

4.1 URAND

URAND produces a sequence of numbers by setting

$$Y_{n+1} = aY_n + C \pmod{m} \quad n \geq 1$$

on the n^{th} call of URAND. These are converted into floating-point numbers in the interval $[0,1)$ and returned as the value of URAND. The resulting value of Y_{n+1} is returned through the parameter IY and used for the actual parameter in the subsequent call. On the first call of URAND, IY should be initialized to an arbitrary integer value. The value of m, a and c are computed automatically upon the initial entry. A different starting value of the variable IY will produce a different sequence of random numbers.

```
      REAL FUNCTION URAND(IY)
      INTEGER IY
C
C        URAND IS A UNIFORM RANDOM NUMBER GENERATOR BASED
C     ON THEORY AND SUGGESTIONS GIVEN IN D.E. KNUTH(1969),
C     VOL 2.  THE INTEGER IY SHOULD BE INITIALIZED TO AN
C     ARBITRARY INTEGER PRIOR TO THE FIRST CALL TO URAND.
C     THE CALLING PROGRAM SHOULD NOT ALTER THE VALUE OF IY
C     BETWEEN SUBSEQUENT CALLS TO URAND.  VALUES OF URAND
C     WILL BE RETURNED IN THE INTERVAL (0,1).
C
         INTEGER IA,IC,ITWO,M2,M,MIC
         DOUBLE PRECISION HALFM
         REAL S
         DOUBLE PRECISION DATAN,DSQRT
         DATA M2/0/,ITWO/2/
         IF (M2.NE.0) GO TO 20
C
C     IF FIRST ENTRY, COMPUTE MACHINE INTEGER WORD LENGTH
C
         M = 1
      10 M2 = M
         M = ITWO*M2
         IF (M.GT.M2) GO TO 10
         HALFM = M2
C
C     COMPUTE MULTIPLIER AND INCREMENT FOR LINEAR CONGRUENTIAL
      METHOD
C
```

```
      IA = 8*IDINT(HALFM*DATAN(1.D0)/8.D0) + 5
      IC = 2*IDINT(HALFM*(0.5D0-DSQRT(3.D0)/6.D0)) + 1
      MIC = (M2 - IC) + M2
C
C   S IS THE SCALE FACTOR FOR CONVERTING TO FLOATING POINT
C
      S = 0.5/HALFM
C
C   COMPUTE NEXT RANDOM NUMBER
C
   20 IY = IY*IA
C
C   THE FOLLOWING STATEMENT IS FOR COMPUTERS WHICH DO NOT
C   ALLOW INTEGER OVERFLOW ON ADDITION
C
      IF (IY.GT.MIC)IY = (IY - M2) - M2
C
      IY = IY + IC
C
C   THE FOLLOWING STATEMENT IS FOR COMPUTERS WHERE THE WORD
C   LENGTH FOR ADDITION IS GREATER THAN FOR MULTIPLICATION
C
      IF (IY/2.GT.M2)IY = (IY - M2) - M2
C
C   THE FOLLOWING STATEMENT IS FOR COMPUTERS WHERE INTEGER
C   OVERFLOW AFFECTS THE SIGN BIT
C
      IF (IY.LT.0)IY = (IY + M2) + M2
      URAND = FLOAT(IY)*S
      RETURN
      END
```

4.2. COMPARISON OF RESULTS

The tests described above are applied to 5 sets of 50,000 digits generated by these two random number generators each. The results are shown in the following tables together with the goodness of fit chi-square for each set.

Tables A-1 to A-10 in Appendix A give data for Frequency and Serial tests, for RANF and URAND respectively. The respective chi-square values are as follows.

TABLE 4.1 FREQUENCY TEST WITH 9 d.f.

RANF		URAND	
Chi-Sq.	Probability	Chi-Sq.	Probability
6.8596	.65	18.2032	.035
6.6812	.67	10.7532	.29
15.8884	.09	7.0972	.63
6.7912	.66	10.6436	.31
7.7036	.57	4.4644	.88

Comparatively RANF is better than URAND.

TABLE 4.2 SERIAL TEST WITH 90 d.f.

RANF		URAND	
Chi-Sq.	Probability	Chi-Sq.	Probability
92.9360	.40	93.9200	.37
145.3680	$<$ .001	103.6360	.156
80.8200	.74	117.1320	.03
71.7320	.92	127.7280	.007
96.5640	.30	93.1760	.39

In RANF one falls below .1 level and one falls above .9; in URAND two fall below .1 level. Both RANF and URAND are about the same.

Tables B-1 and B-2 in appendix B give data for the Poker test, for RANF and URAND respectively.

The Chi-square values with 6 d.f. are given below in Table 4.3.

TABLE 4.3 POKER TEST WITH 6 d.f.

RANF		URAND	
Chi-Sq.	Probability	Chi-Sq.	Probability
3.57434	.74	5.25390	.52
3.40403	.76	10.06237	.13
2.55351	.86	7.25589	.30
5.93730	.44	5.53664	.48
6.30317	.40	4.32526	.65

Tables C-1 to C-5 in appendix C provides data for the Gap test. Table 4.4 gives the chi-square values based on these data for RANF and URAND, with 16 d.f.

TABLE 4.4 GAP TEST WITH 16 d.f.

RANF		URAND	
Chi-Sq.	Probability	Chi-Sq.	Probability
13.3688	.64	25.1960	.07
17.4315	.37	19.6205	.23
9.0531	.91	10.7083	.83
21.6936	.16	14.4559	.563
18.6552	.29	19.0242	.27

Both Poker and Gap test show that RANF is slightly better than URAND.

Tables 4.5a and 4.5b provide data for the RUN test, for RANF and URAND respectively. Five sets of 10,000 numbers each is used to compute observed frequencies. The expected frequencies are calculated using formula given in 3.2.

TABLE 4.5a RUN TEST - RANF

RUN LENGTH	EXP. FREQ.	OBS. FREQ. 1	OBS. FREQ. 2	OBS. FREQ. 3	OBS. FREQ. 4	OBS. FREQ. 5
1	4166.75	4061	4201	4287	4121	4089
2	1833.10	1845	1821	1838	1867	1819
3	527.65	541	534	520	531	518
4	115.04	120	112	117	125	128
5	20.33	20	27	16	17	24
≥ 6	3.47	3	5	5	4	2

TABLE 4.5b RUN TEST - URAND

RUN LENGTH	EXP. FREQ.	OBS. FREQ. 1	OBS. FREQ. 2	OBS. FREQ. 3	OBS. FREQ. 4	OBS. FREQ. 5
1	4166.75	4200	4151	4289	4001	4019
2	1833.10	1821	1853	1827	1830	1867
3	527.65	530	550	518	523	509
4	115.04	126	112	109	121	120
5	20.33	20	26	19	25	16
≥ 6	3.47	3	4	4	5	5

The above tables show obviously satisfactory results for both RANF and URAND.

CHAPTER V - CONCLUSIONS

Five tests (Frequency, Serial, Poker, Gap and Run test) have been applied to the digits of random numbers generated by RANF and URAND respectively to test the randomness of the numbers and goodness of fit chi-square statistic are computed with appropriate degrees of freedom. Both RANF and URAND random number generator have passed these tests satisfactorily, but the chi-square tests show that RANF is slightly better than URAND; although the difference is not highly significant.

Bibliography

[1] Forsythe, G.E., Michael, M.A. and Moler, C.B., "Computer methods for mathematical computations" Prentice-Hall, Inc.

[2] Green, B.F., Smith, Jr. and Klem, L., "Empirical tests of an additive Random number generator", JACM, 6 (1959), pp. 527-537.

[3] IBM. "Data processing techniques - Random number generation and testing". IBM branch offices, 1959.

[4] Kendall, M.G. and Smith, B.B., "Randomness and Random sampling numbers". Journal of Royal Stat. Soc. A, 101 (1938), pp. 147-166.

[5] Kendall, M.G. and Smith, B.B., "Second paper on Random sampling numbers", Journal of Royal Stat. Soc. (supplement) B, 6 (1939), pp. 51-61.

[6] Knuth, D.E., "Seminumerical algorithms - The art of computer programming", Vol. 2, Addison-Wesley, Chapter 3, 1969.

[7] Landis, J.R. and Feinstein, A.R., "An empirical comparison of random number acquired by computer generation and from Rand tables", Computers and Biomedical Research, 6 (1973), pp. 322-326.

[8] Marsaglia, G. and Bray, T.A., "On-line random number generators and their use in combinations", ACM Comm., 11 (1968), pp. 757-759.

[9] Nance, R.E. and Overstreat, C., "Bibliography 29". Computing Reviews, 13 (1972), pp. 498-508.

[10] Schrage, L., "A more portable FORTRAN random number generator", ACM Trans. Math. Software, 5 (1979) pp. 132-138.

[11] The RAND Corporation, "A million Random digits with 100,000 Normal deviates", The Free Press, New York.

[12] Yven, C.K., "Testing random number generators by Walsh transform", IEEE Trans. Computers, C-26 (1977), pp. 329-333.

APPENDIX A

Data for Frequency and Serial Tests - RANF and URAND

FIRST DIGIT	SECOND DIGIT 0	1	2	3	4	5	6	7	8	9	ROW TOTAL
0	502	478	512	483	469	490	517	503	526	479	4959
1	476	515	492	475	531	503	503	495	471	475	4936
2	529	492	508	505	523	486	543	469	50	473	5030
3	480	487	513	496	458	507	513	476	511	475	4916
4	514	506	513	485	535	540	479	498	505	517	5092
5	527	462	532	477	505	495	506	507	524	506	5041
6	490	525	486	524	489	506	484	482	483	544	5013
7	479	460	484	499	506	501	487	499	512	500	4927
8	486	473	478	517	524	532	505	484	521	544	5064
9	476	538	512	455	552	481	477	514	508	509	5022
COL. TOTAL	4959	4936	5030	4916	5092	5041	5013	4927	5064	5022	50,000

TABLE A-1 DATA SET 1 - FOR FREQUENCY AND SERIAL TESTS - RANF

SECOND DIGIT

FIRST DIGIT	0	1	2	3	4	5	6	7	8	9	ROW TOTAL
0	537	539	496	513	517	484	497	474	526	490	5073
1	482	481	488	544	509	497	511	456	501	477	4916
2	502	452	471	501	513	460	458	483	491	529	4860
3	527	509	537	483	498	531	532	467	505	459	5048
4	484	505	518	509	520	540	492	485	555	539	5147
5	474	513	504	504	506	489	459	487	535	500	4971
6	536	467	433	549	497	485	500	515	471	500	4953
7	505	491	465	466	542	509	446	539	501	473	4937
8	522	515	487	490	544	469	533	505	532	528	5125
9	504	444	461	489	501	507	525	526	508	505	4970
COL. TOTAL	5073	4916	4860	5048	5147	4971	4953	4937	5125	4970	50000

TABLE A-2 DATA SET 2 - FOR FREQUENCY AND SERIA TEST - RANF

FIRST DIGIT	SECOND DIGIT										
	0	1	2	3	4	5	6	7	8	9	ROW TOTAL
0	473	506	457	538	500	491	510	496	489	490	4950
1	519	473	512	497	493	471	483	480	482	528	4938
2	450	502	515	504	483	499	520	470	479	522	4944
3	500	523	465	510	542	472	503	557	497	513	5082
4	507	471	517	507	513	492	526	481	549	529	5092
5	489	500	493	492	520	497	505	516	457	491	4960
6	514	494	488	497	500	509	503	493	514	500	5012
7	518	474	501	530	486	490	498	491	510	490	4988
8	481	495	510	500	490	517	479	489	489	511	4961
9	499	500	486	507	565	522	485	515	495	499	5073
COL. TOTAL	4950	4938	4944	5082	5092	4960	5012	4988	4961	5073	50000

TABLE A-3 DATA SET 3 - FOR FREQUENCY AND SERIAL TEST - RANF

FIRST DIGIT	SECOND DIGIT 0	1	2	3	4	5	6	7	8	9	ROW TOTAL
0	543	503	483	541	508	493	497	520	490	496	5074
1	520	520	498	489	473	499	497	508	506	520	5030
2	508	556	475	481	477	490	506	489	490	504	4976
3	496	487	522	485	480	536	507	516	490	506	5025
4	520	502	483	496	484	458	505	470	464	497	4879
5	476	479	495	530	493	549	520	513	523	476	5054
6	508	498	486	498	486	513	508	590	497	522	5006
7	503	498	521	488	484	500	469	497	513	517	4990
8	487	494	482	479	503	499	513	501	460	500	4918
9	513	493	531	538	491	517	484	486	485	510	5048
COL. TOTAL	5074	5030	4976	5025	4879	5054	5006	4990	4918	5048	50000

TABLE A-4 DATA SET 4 - FOR FREQUENCY AND SERIAL TEST - RANF

		SECOND DIGIT									
FIRST DIGIT		0	1	2	3	4	5	6	7	8	9	ROW TOTAL
	0	505	485	484	484	492	504	509	492	489	516	4960
	1	500	467	469	472	525	501	468	482	503	477	4864
	2	493	519	464	490	477	508	512	513	533	505	5014
	3	488	507	528	487	545	493	451	505	525	508	5037
	4	517	468	529	524	483	532	513	516	478	497	5057
	5	530	492	532	493	488	489	461	501	507	530	5023
	6	466	479	495	508	532	505	493	463	507	515	4963
	7	458	453	494	513	502	490	547	475	556	486	4974
	8	485	484	518	531	523	520	520	519	504	501	5105
	9	518	510	501	535	490	481	489	508	503	468	5003
COL. TOTAL		4960	4864	5014	5037	5057	5023	4963	4974	5105	5003	50000

TABLE A-5 DATA SET 5 - FOR FREQUENCY AND SERIAL TEST - RANF

SECOND DIGIT

FIRST DIGIT	0	1	2	3	4	5	6	7	8	9	ROW TOTAL
0	507	450	511	492	466	484	537	528	511	514	5000
1	497	538	496	485	501	505	494	491	506	480	4993
2	533	504	548	528	512	491	503	527	497	505	5148
3	479	473	492	507	481	521	505	519	509	533	5019
4	517	501	509	495	477	472	489	470	465	491	4886
5	483	518	518	480	484	511	431	506	519	501	4951
6	466	509	511	490	474	488	490	472	507	517	4924
7	502	501	544	489	496	477	478	480	521	487	4975
8	506	499	493	551	482	460	512	489	477	540	5009
9	510	500	526	502	513	542	485	493	497	527	5095
COL. TOTAL	5000	4993	5148	5019	4886	4951	4924	4975	5009	5095	50000

TABLE A-6 DATA SET 1 - FOR FREQUENCY AND SERIAL TEST - URAND

FIRST DIGIT	SECOND DIGIT 0	1	2	3	4	5	6	7	8	9	ROW TOTAL
0	493	485	497	540	516	494	463	523	508	519	5038
1	525	505	501	497	509	514	519	490	480	526	5066
2	527	535	487	501	551	505	472	515	459	469	5021
3	466	487	498	498	496	512	480	504	484	487	4912
4	487	494	535	507	494	505	514	525	529	475	5065
5	518	543	479	484	483	547	440	511	523	520	5048
6	506	532	509	484	483	484	520	447	495	480	4940
7	508	524	526	454	549	464	533	508	492	488	5046
8	502	474	480	484	497	506	506	495	496	490	4930
9	506	487	509	463	487	517	493	528	464	480	4934
COL. TOTAL	5038	5066	5021	4912	5065	5048	4940	5046	4930	4934	50000

TABLE A-7 DATA SET 2 - FOR FREQUENCY AND SERIAL TEST - URAND

SECOND DIGIT

FIRST DIGIT	0	1	2	3	4	5	6	7	8	9	ROW TOTAL
0	502	535	519	529	490	504	498	518	543	513	5151
1	482	473	492	522	482	497	517	478	468	481	4892
2	487	463	501	495	509	493	497	491	489	523	4948
3	507	500	506	488	482	514	503	469	536	502	5007
4	518	503	468	450	464	509	518	503	474	518	4925
5	546	468	489	545	462	498	483	487	536	490	5004
6	569	501	500	488	531	510	504	462	490	510	5065
7	484	476	489	500	502	457	473	490	520	557	4948
8	509	520	496	473	505	500	548	503	463	480	4997
9	547	453	488	517	498	522	524	547	478	489	5063
COL. TOTAL	5151	4892	4948	5007	4925	5004	5065	4948	4997	5063	50000

TABLE A-8 DATA SET 3 - FOR FREQUENCY AND SERIAL TEST - URAND

SECOND DIGIT

FIRST DIGIT	0	1	2	3	4	5	6	7	8	9	ROW TOTAL
0	499	520	472	497	547	507	539	499	504	561	5145
1	513	515	508	427	533	534	518	503	498	483	5032
2	519	481	467	528	487	505	511	506	491	461	4956
3	522	449	465	418	492	488	468	488	499	488	4777
4	513	521	500	485	505	482	508	524	510	481	5029
5	525	512	540	458	518	500	518	485	468	530	5054
6	515	511	532	451	477	518	529	510	531	517	5091
7	536	497	479	523	495	528	467	464	466	481	4936
8	522	500	506	475	496	513	497	467	495	497	4968
9	481	526	487	515	479	479	536	490	506	513	5012
COL. TOTAL	5145	5032	4956	4777	5029	5054	5091	4936	4968	5012	50000

TABLE A-9 DATA SET 4 - FOR FREQUENCY AND SERIAL TEST - URAND

SECOND DIGIT

FIRST DIGIT	0	1	2	3	4	5	6	7	8	9	ROW TOTAL
0	522	496	532	505	496	514	514	478	512	532	5101
1	509	478	509	493	527	540	500	496	498	478	5028
2	536	490	502	477	502	477	408	513	467	494	4966
3	506	503	517	490	496	489	456	495	515	494	4961
4	511	525	521	533	440	407	521	500	459	490	5007
5	519	518	467	494	528	512	509	486	508	500	5041
6	553	490	480	485	477	476	497	481	506	518	4963
7	476	481	490	519	512	505	458	501	516	531	4989
8	512	523	462	479	503	478	513	519	461	478	4928
9	457	524	486	486	526	543	487	520	486	501	5016
COL. TOTAL	5101	5028	4966	4961	5007	5041	4963	4989	4928	5016	50000

TABLE A-10 DATA SET 5 - FOR FREQUENCY AND SERIAL TEST - URAND

APPENDIX B

Data for Poker Test - RANF and URAND

COMBINATION	EXP. FREQ.	OBS. FREQ. 1	OBS. FREQ. 2	OBS. FREQ. 3	OBS. FREQ. 4	OBS. FREQ. 5
abcde	3024.0	3022	2977	3040	3078	3045
aabcd	5040.0	5080	5029	5042	5063	4967
aabbc	1080.0	1032	1094	1092	1050	1086
aaabc	720.0	724	753	701	683	744
aaabb	90.0	99	96	85	80	104
aaaab	45.0	42	50	40	44	52
aaaaa	1.0	1	1	0	2	2

TABLE B-1 DATA FOR POKER TEST - RANF

COMBINATION	EXP. FREQ.	OBS. FREQ. 1	OBS. FREQ. 2	OBS. FREQ. 3	OBS. FREQ. 4	OBS. FREQ. 5
abcbe	3024.0	3046	2996	3085	2992	3016
aabcd	5040.0	5073	4979	5009	5094	5008
aabbc	1080.0	1080	1144	1056	1095	1073
aaabc	720.0	668	722	721	704	752
aaabb	90.0	87	106	97	75	97
aaaab	45.0	44	53	32	40	52
aaaaa	1.0	2	0	0	0	2

TABLE B-2 DATA FOR POKER TEST - URAND

APPENDIX C

Data for Gap Test - RANF and URAND

TABLE C-1 DATA SET 1 - FOR GAP TEST - RANF and URAND

GAP	RANF		URAND	
LENGTH	EXP. FREQ.	OBS. FREQ.	EXP. FREQ.	OBS. FREQ.
0	495.00	464	503.80	523
1	445.50	440	453.42	498
2	400.95	401	408.08	409
3	360.85	384	367.27	331
4	324.77	343	330.54	331
5	292.29	293	297.49	288
6	263.06	272	267.74	264
7	236.76	248	240.97	239
8	213.08	208	216.87	224
9	191.77	187	195.18	226
10	172.60	166	175.66	158
11	155.34	141	158.10	161
12	139.80	138	142.29	130
13	125.82	107	128.06	115
14	113.24	128	115.25	120
15	101.92	114	103.73	98
≥16	917.24	916	933.55	923

TABLE C-2 DATA SET 2 - FOR GAP TEST - RANF and URAND

GAP LENGTH	RANF EXP. FREQ.	RANF OBS. FREQ.	URAND EXP. FREQ.	URAND OBS. FREQ
0	507.40	540	510.10	534
1	456.66	424	459.09	462
2	410.99	399	413.18	395
3	369.89	387	371.86	364
4	332.91	353	334.68	334
5	299.61	318	301.21	299
6	269.65	274	271.09	298
7	242.69	261	243.98	225
8	218.42	203	219.58	237
9	196.58	190	197.62	210
10	176.92	191	177.86	182
11	159.23	145	160.07	147
12	143.30	126	144.07	146
13	128.97	122	129.66	138
14	116.08	102	116.69	100
15	104.47	100	105.03	115
≥16	940.22	939	945.23	915

TABLE C-3 DATA SET 3 - FOR GAP TEST - RANF and URAND

GAP LENGTH	RANF EXP. FREQ.	RANF OBS. FREQ.	URAND EXP. FREQ.	URAND OBS. FREQ.
0	495.90	512	500.00	494
1	446.31	414	450.00	443
2	401.68	399	405.00	426
3	361.51	375	364.50	347
4	325.36	321	328.05	304
5	292.82	293	295.24	305
6	263.54	259	265.72	265
7	237.19	245	239.15	227
8	213.47	201	215.23	210
9	192.12	205	193.71	215
10	172.91	165	174.34	163
11	155.62	164	156.91	167
12	140.06	136	141.21	138
13	126.05	123	127.09	142
14	113.45	105	114.38	118
15	102.10	92	102.95	106
≥16	918.91	950	926.51	930

TABLE C-4 DATA SET 4 - FOR GAP TEST - RANF and URAND

GAP LENGTH	RANF EXP. FREQ.	RANF OBS. FREQ.	URAND EXP. FREQ.	URAND OBS. FREQ.
0	496.00	492	514.50	502
1	446.40	411	463.05	472
2	401.76	413	416.74	444
3	361.58	383	395.07	387
4	325.43	299	337.56	358
5	292.38	335	303.81	319
6	263.59	233	273.43	236
7	237.24	227	246.08	251
8	213.51	228	221.48	240
9	192.16	191	199.33	205
10	172.94	169	179.40	194
11	155.65	152	161.46	165
12	140.09	159	145.31	140
13	126.08	116	130.78	125
14	113.47	118	117.70	115
15	102.12	99	105.93	113
≥16	919.10	935	953.38	879

TABLE C-5 DATA SET 5 - FOR GAP TEST - RANF and URAND

GAP LENGTH	RANF EXP. FREQ.	RANF OBS. FREQ.	URAND EXP. FREQ.	URAND OBS. FREQ.
0	507.30	512	515.10	496
1	456.57	436	463.59	475
2	410.91	477	417.23	452
3	369.82	372	375.51	372
4	332.84	316	337.96	319
5	299.56	299	304.16	300
6	269.60	267	273.75	311
7	242.64	234	246.37	243
8	218.38	201	221.73	241
9	196.54	188	199.56	193
10	176.88	180	179.60	171
11	159.20	157	161.64	174
12	143.28	152	145.48	162
13	128.95	139	130.93	136
14	116.05	121	117.84	128
15	104.45	112	106.05	113
≥16	940.04	901	954.49	865

VITA

Candidate's full name: Kanchan Kumar

Place and date of birth: Allahabad, U.P., India
October 1, 1954

Permanent address: c/o Dr. B. K. Agarwal
14-A, Bank Road
Allahabad, U.P., India

Schools attended: D.P. Girls Inter College
(with dates) Allahabad, India
(1964-1971)

Universities attended: Allahabad University
(with dates and Allahabad, India
degrees obtained) (1971-1975)
B.Sc.

University of New Brunswick
N.B., Canada
(1975-1979)
M.A. - Statistic

Lehigh University
Pa., U.S.A.
(1978-1980)