"XRNG" RANDOM NUMBER GENERATOR

PROJECT TEST REPORT

Andrew Vincze
Marc Gallo

RNG Research
40 Franklin Street
New London, CT 06320
(860) 444-2996
www.rngresearch.com
December 12, 2000

ABSTRACT: Empirical evidence of XRNG method is presented. XRNG passes all tests for randomness specified by Federal Information Processing Standards Publication 140-1 and Diehard test battery.

rng RESEARCH

1.0 Introduction

This document provides test result data for the XRNG true-random number generation method to demonstrate empirical evidence of the method to interested parties outside RNG Research. The scope of this document is limited to Federal Information Processing Standards Publication 140-1 (FIPS PUB 140-1)[1] and Diehard[2] test results. XRNG apparatus have passed all tests for randomness to which they have been subjected.

2.0 Background

XRNG is a robust method of generating truly random sequences of numbers which fall into the discrete uniform distribution over the desired interval. This is accomplished by measuring the output of a truly random noise source (e.g., semiconductor noise) with an analog-to-digital (A/D) converter, and applying a modular reduction to the measurements. Assuming the normal distribution for the noise, the theory of operation predicts maximum biases (expressed as the difference between the computed probability of an XRNG outputting a given number and the probability based on an ideal uniform distribution) for two sets of conditions as follows:

A/D Converter Resolution	Standard Deviation of A/D Converter Output Distribution	Reduction Modulus	Theoretical Bias is less than
16 bits	4,096 codes	256	3×10^{-12}
8 bits	16 codes	16	4×10^{-9}

3.0 Procedure

Truly random numbers modulo-256 were generated by XRNG apparatus comprising a reverse-biased P-N junction semiconductor noise source, 16-bit A/D converter, and interface means by which data bits D8 through D15 were discarded. Four (4) files of 32,768 bytes each were generated and the first 20,000 bits (i.e., the first 2,500 bytes) of each were subjected to the tests for randomness specified in FIPS PUB 140-1. Eight (8) files of 10,485,760 bytes each were also generated and subjected to all fifteen (15) tests in the MS-DOS version of the Diehard test suite. The first 20,000 bits of each of these files were also subjected to the FIPS PUB 140-1 tests as above.

Lastly, random numbers modulo-16 were generated by XRNG apparatus comprising a reverse-biased P-N junction semiconductor noise source, 8-bit A/D converter and interface means by which data bits D4 through D7 were discarded. Odd nybbles in a random sequence of 20,971,520 nybbles were concatenated with the immediately-following even nybble to form a sequence of 10,485,760 bytes, which was then subjected to the Diehard test suite.

4.0 Results

XRNG apparatus successfully passed all tests for randomness. No deviation from the ideal uniform distribution was detected in the XRNG data.

Test results are tabulated as follows:

TESTS FOR RANDOMNESS
ON RANDOM SEQUENCES GEREATED BY XRNG METHOD

file	A/D Converter Resolution	Standard Deviation of A/D Converter Output Distribution	Reduction Modulus	Theoretical Bias is less than	Tests Run F=FIPS D=Diehard	Observed Bias
sample1.rng (32K)	16 bits	4,300 codes	256	3×10^{-12}	F	none
sample2.rng (32K)	16 bits	4,300 codes	256	3×10^{-12}	F	none
sample3.rng (32K)	16 bits	4,300 codes	256	3×10^{-12}	F	none
sample4.rng (32K)	16 bits	4,300 codes	256	3×10^{-12}	F	none
block0.rng (10M)	16 bits	4,300 codes	256	3×10^{-12}	F,D	none
block1.rng (10M)	16 bits	4,300 codes	256	3×10^{-12}	F,D	none
block2.rng (10M)	16 bits	4,300 codes	256	3×10^{-12}	F,D	none
block3.rng (10M)	16 bits	4,300 codes	256	3×10^{-12}	F,D	none
block4.rng (10M)	16 bits	4,300 codes	256	3×10^{-12}	F,D	none
block5.rng (10M)	16 bits	4,300 codes	256	3×10^{-12}	F,D	none
block6.rng (10M)	16 bits	4,300 codes	256	3×10^{-12}	F,D	none
block7.rng (10M)	16 bits	4,300 codes	256	3×10^{-12}	F,D	none
blockx.rng (10M)	8 bits	16 codes	16	4×10^{-9}	D	none

Output reports generated by the program running the FIPS PUB 140-1 tests are presented in Appendix A. The full text of the Diehard output files are included in Appendix B.

5.0 Conclusions

Data produced by XRNG method has passed all tests for randomness to which it has been subjected. No bias has been detected. Empirical evidence of the method is thus presented.

6.0 References

1. United States Government Commerce Department, Ronald H. Brown, Secretary. National Institute of Standards and Technology. (1994) Federal Information Processing Standards Publication 140-1: Security Requirements for Cryptographic Modules (see http://csrc.ncsl.nist.gov/fips/fips1401.htm)

2. Marsaglia, G. (1997) Diehard: a battery of tests for random number generators. published online on Florida State University Department of Statistics web site at http://stat.fsu.edu/~geo/diehard.html.

Appendix A: FIPS PUB 140-1 Test Result Details

```
Results of FIPS 140-1 Specified Tests on sample1.rng

BLOCK 1
     Monobit X= 9936           PASS (pass if 9654 < X < 10346)
     Poker X= 9.7728           PASS (pass if 1.03 < X < 57.4)
     Run length  Zeros  Ones
           1      2536  2547  pass (pass if 2267 < X < 2733)
           2      1255  1258  pass (pass if 1079 < X < 1421)
           3       597   640  pass (pass if 502 < X < 748)
           4       336   287  pass (pass if 223 < X < 402)
           5       147   157  pass (pass if 90 < X < 223)
           6+      165   147  pass (pass if 90 < X < 223)
                              PASS (pass if all twelve counts pass)
     No long run (LEN => 34)  PASS (pass if no long run)

BLOCK 2
     Monobit X= 10066          PASS (pass if 9654 < X < 10346)
     Poker X= 9.0048           PASS (pass if 1.03 < X < 57.4)
     Run length  Zeros  Ones
           1      2481  2452  pass (pass if 2267 < X < 2733)
           2      1254  1205  pass (pass if 1079 < X < 1421)
           3       616   670  pass (pass if 502 < X < 748)
           4       314   315  pass (pass if 223 < X < 402)
           5       153   175  pass (pass if 90 < X < 223)
           6+      153   153  pass (pass if 90 < X < 223)
                              PASS (pass if all twelve counts pass)
     No long run (LEN => 34)  PASS (pass if no long run)

BLOCK 3
     Monobit X= 10032          PASS (pass if 9654 < X < 10346)
     Poker X= 16.0832          PASS (pass if 1.03 < X < 57.4)
     Run length  Zeros  Ones
           1      2444  2490  pass (pass if 2267 < X < 2733)
           2      1268  1220  pass (pass if 1079 < X < 1421)
           3       639   613  pass (pass if 502 < X < 748)
           4       312   310  pass (pass if 223 < X < 402)
           5       144   173  pass (pass if 90 < X < 223)
           6+      160   160  pass (pass if 90 < X < 223)
                              PASS (pass if all twelve counts pass)
     No long run (LEN => 34)  PASS (pass if no long run)

BLOCK 4
     Monobit X= 9957           PASS (pass if 9654 < X < 10346)
     Poker X= 9.6384           PASS (pass if 1.03 < X < 57.4)
     Run length  Zeros  Ones
           1      2487  2547  pass (pass if 2267 < X < 2733)
           2      1267  1228  pass (pass if 1079 < X < 1421)
           3       590   601  pass (pass if 502 < X < 748)
           4       356   291  pass (pass if 223 < X < 402)
           5       161   183  pass (pass if 90 < X < 223)
           6+      139   151  pass (pass if 90 < X < 223)
                              PASS (pass if all twelve counts pass)
     No long run (LEN => 34)  PASS (pass if no long run)
```

```
BLOCK 5
     Monobit X= 10134          PASS (pass if 9654 < X < 10346)
     Poker X= 22.3424          PASS (pass if 1.03 < X < 57.4)
     Run length  Zeros  Ones
           1      2483  2439  pass (pass if 2267 < X < 2733)
           2      1240  1247  pass (pass if 1079 < X < 1421)
           3       624   588  pass (pass if 502 < X < 748)
           4       299   326  pass (pass if 223 < X < 402)
           5       148   169  pass (pass if 90 < X < 223)
           6+      161   186  pass (pass if 90 < X < 223)
                              PASS (pass if all twelve counts pass)
     No long run (LEN => 34)  PASS (pass if no long run)



Results of FIPS 140-1 Specified Tests on sample2.rng

BLOCK 1
     Monobit X= 10010          PASS (pass if 9654 < X < 10346)
     Poker X= 11.008           PASS (pass if 1.03 < X < 57.4)
     Run length  Zeros  Ones
           1      2519  2530  pass (pass if 2267 < X < 2733)
           2      1210  1218  pass (pass if 1079 < X < 1421)
           3       651   644  pass (pass if 502 < X < 748)
           4       322   308  pass (pass if 223 < X < 402)
           5       158   138  pass (pass if 90 < X < 223)
           6+      145   166  pass (pass if 90 < X < 223)
                              PASS (pass if all twelve counts pass)
     No long run (LEN => 34)  PASS (pass if no long run)

BLOCK 2
     Monobit X= 9998           PASS (pass if 9654 < X < 10346)
     Poker X= 11.6928          PASS (pass if 1.03 < X < 57.4)
     Run length  Zeros  Ones
           1      2479  2476  pass (pass if 2267 < X < 2733)
           2      1255  1274  pass (pass if 1079 < X < 1421)
           3       654   636  pass (pass if 502 < X < 748)
           4       321   325  pass (pass if 223 < X < 402)
           5       136   139  pass (pass if 90 < X < 223)
           6+      159   155  pass (pass if 90 < X < 223)
                              PASS (pass if all twelve counts pass)
     No long run (LEN => 34)  PASS (pass if no long run)

BLOCK 3
     Monobit X= 10003          PASS (pass if 9654 < X < 10346)
     Poker X= 19.4944          PASS (pass if 1.03 < X < 57.4)
     Run length  Zeros  Ones
           1      2497  2502  pass (pass if 2267 < X < 2733)
           2      1314  1247  pass (pass if 1079 < X < 1421)
           3       584   638  pass (pass if 502 < X < 748)
           4       282   312  pass (pass if 223 < X < 402)
           5       165   151  pass (pass if 90 < X < 223)
           6+      164   157  pass (pass if 90 < X < 223)
                              PASS (pass if all twelve counts pass)
     No long run (LEN => 34)  PASS (pass if no long run)
```

```
BLOCK 4
     Monobit X= 9832           PASS (pass if 9654 < X < 10346)
     Poker X= 16.7616          PASS (pass if 1.03 < X < 57.4)
     Run length  Zeros  Ones
           1      2399  2486  pass (pass if 2267 < X < 2733)
           2      1201  1219  pass (pass if 1079 < X < 1421)
           3       676   639  pass (pass if 502 < X < 748)
           4       321   313  pass (pass if 223 < X < 402)
           5       176   130  pass (pass if 90 < X < 223)
           6+      169   154  pass (pass if 90 < X < 223)
                              PASS (pass if all twelve counts pass)
     No long run (LEN => 34)  PASS (pass if no long run)

BLOCK 5
     Monobit X= 9915           PASS (pass if 9654 < X < 10346)
     Poker X= 9.664            PASS (pass if 1.03 < X < 57.4)
     Run length  Zeros  Ones
           1      2437  2471  pass (pass if 2267 < X < 2733)
           2      1251  1255  pass (pass if 1079 < X < 1421)
           3       627   633  pass (pass if 502 < X < 748)
           4       313   327  pass (pass if 223 < X < 402)
           5       178   125  pass (pass if 90 < X < 223)
           6+      163   158  pass (pass if 90 < X < 223)
                              PASS (pass if all twelve counts pass)
     No long run (LEN => 34)  PASS (pass if no long run)



Results of FIPS 140-1 Specified Tests on sample3.rng

BLOCK 1
     Monobit X= 9973           PASS (pass if 9654 < X < 10346)
     Poker X= 14.4512          PASS (pass if 1.03 < X < 57.4)
     Run length  Zeros  Ones
           1      2496  2447  pass (pass if 2267 < X < 2733)
           2      1218  1288  pass (pass if 1079 < X < 1421)
           3       640   632  pass (pass if 502 < X < 748)
           4       317   316  pass (pass if 223 < X < 402)
           5       151   162  pass (pass if 90 < X < 223)
           6+      164   141  pass (pass if 90 < X < 223)
                              PASS (pass if all twelve counts pass)
     No long run (LEN => 34)  PASS (pass if no long run)

BLOCK 2
     Monobit X= 10003          PASS (pass if 9654 < X < 10346)
     Poker X= 20.8576          PASS (pass if 1.03 < X < 57.4)
     Run length  Zeros  Ones
           1      2538  2588  pass (pass if 2267 < X < 2733)
           2      1273  1239  pass (pass if 1079 < X < 1421)
           3       643   619  pass (pass if 502 < X < 748)
           4       311   293  pass (pass if 223 < X < 402)
           5       146   162  pass (pass if 90 < X < 223)
           6+      146   155  pass (pass if 90 < X < 223)
                              PASS (pass if all twelve counts pass)
     No long run (LEN => 34)  PASS (pass if no long run)
```

```
BLOCK 3
     Monobit X= 9962          PASS (pass if 9654 < X < 10346)
     Poker X= 17.568          PASS (pass if 1.03 < X < 57.4)
     Run length  Zeros  Ones
           1      2535  2601  pass (pass if 2267 < X < 2733)
           2      1264  1245  pass (pass if 1079 < X < 1421)
           3       626   618  pass (pass if 502 < X < 748)
           4       339   296  pass (pass if 223 < X < 402)
           5       148   129  pass (pass if 90 < X < 223)
           6+      144   167  pass (pass if 90 < X < 223)
                              PASS (pass if all twelve counts pass)
     No long run (LEN => 34)  PASS (pass if no long run)

BLOCK 4
     Monobit X= 10055         PASS (pass if 9654 < X < 10346)
     Poker X= 18.592          PASS (pass if 1.03 < X < 57.4)
     Run length  Zeros  Ones
           1      2490  2492  pass (pass if 2267 < X < 2733)
           2      1220  1181  pass (pass if 1079 < X < 1421)
           3       627   670  pass (pass if 502 < X < 748)
           4       346   300  pass (pass if 223 < X < 402)
           5       147   165  pass (pass if 90 < X < 223)
           6+      148   170  pass (pass if 90 < X < 223)
                              PASS (pass if all twelve counts pass)
     No long run (LEN => 34)  PASS (pass if no long run)

BLOCK 5
     Monobit X= 9910          PASS (pass if 9654 < X < 10346)
     Poker X= 11.296          PASS (pass if 1.03 < X < 57.4)
     Run length  Zeros  Ones
           1      2445  2541  pass (pass if 2267 < X < 2733)
           2      1251  1212  pass (pass if 1079 < X < 1421)
           3       649   603  pass (pass if 502 < X < 748)
           4       320   307  pass (pass if 223 < X < 402)
           5       140   154  pass (pass if 90 < X < 223)
           6+      176   165  pass (pass if 90 < X < 223)
                              PASS (pass if all twelve counts pass)
     No long run (LEN => 34)  PASS (pass if no long run)



Results of FIPS 140-1 Specified Tests on sample4.rng

BLOCK 1
     Monobit X= 9962          PASS (pass if 9654 < X < 10346)
     Poker X= 20.096          PASS (pass if 1.03 < X < 57.4)
     Run length  Zeros  Ones
           1      2399  2451  pass (pass if 2267 < X < 2733)
           2      1255  1235  pass (pass if 1079 < X < 1421)
           3       638   608  pass (pass if 502 < X < 748)
           4       325   322  pass (pass if 223 < X < 402)
           5       142   159  pass (pass if 90 < X < 223)
           6+      172   157  pass (pass if 90 < X < 223)
                              PASS (pass if all twelve counts pass)
     No long run (LEN => 34)  PASS (pass if no long run)
```

```
BLOCK 2
     Monobit X= 9957          PASS (pass if 9654 < X < 10346)
     Poker X= 10.88           PASS (pass if 1.03 < X < 57.4)
     Run length  Zeros  Ones
           1      2514  2522  pass (pass if 2267 < X < 2733)
           2      1260  1243  pass (pass if 1079 < X < 1421)
           3       593   621  pass (pass if 502 < X < 748)
           4       306   320  pass (pass if 223 < X < 402)
           5       165   155  pass (pass if 90 < X < 223)
           6+      173   149  pass (pass if 90 < X < 223)
                              PASS (pass if all twelve counts pass)
     No long run (LEN => 34)  PASS (pass if no long run)

BLOCK 3
     Monobit X= 9949          PASS (pass if 9654 < X < 10346)
     Poker X= 18.2784         PASS (pass if 1.03 < X < 57.4)
     Run length  Zeros  Ones
           1      2536  2532  pass (pass if 2267 < X < 2733)
           2      1268  1298  pass (pass if 1079 < X < 1421)
           3       640   623  pass (pass if 502 < X < 748)
           4       279   291  pass (pass if 223 < X < 402)
           5       156   157  pass (pass if 90 < X < 223)
           6+      166   144  pass (pass if 90 < X < 223)
                              PASS (pass if all twelve counts pass)
     No long run (LEN => 34)  PASS (pass if no long run)

BLOCK 4
     Monobit X= 9950          PASS (pass if 9654 < X < 10346)
     Poker X= 12.0192         PASS (pass if 1.03 < X < 57.4)
     Run length  Zeros  Ones
           1      2484  2471  pass (pass if 2267 < X < 2733)
           2      1256  1276  pass (pass if 1079 < X < 1421)
           3       618   657  pass (pass if 502 < X < 748)
           4       304   290  pass (pass if 223 < X < 402)
           5       161   146  pass (pass if 90 < X < 223)
           6+      167   150  pass (pass if 90 < X < 223)
                              PASS (pass if all twelve counts pass)
     No long run (LEN => 34)  PASS (pass if no long run)

BLOCK 5
     Monobit X= 10018         PASS (pass if 9654 < X < 10346)
     Poker X= 21.728          PASS (pass if 1.03 < X < 57.4)
     Run length  Zeros  Ones
           1      2586  2481  pass (pass if 2267 < X < 2733)
           2      1208  1279  pass (pass if 1079 < X < 1421)
           3       589   665  pass (pass if 502 < X < 748)
           4       313   304  pass (pass if 223 < X < 402)
           5       171   154  pass (pass if 90 < X < 223)
           6+      160   144  pass (pass if 90 < X < 223)
                              PASS (pass if all twelve counts pass)
     No long run (LEN => 34)  PASS (pass if no long run)
```

```
Results of FIPS 140-1 Specified Tests on block0.rng

BLOCK 1
     Monobit X= 9915          PASS (pass if 9654 < X < 10346)
     Poker X= 13.664          PASS (pass if 1.03 < X < 57.4)
     Run length  Zeros  Ones
           1      2506  2558  pass (pass if 2267 < X < 2733)
           2      1239  1245  pass (pass if 1079 < X < 1421)
           3       644   629  pass (pass if 502 < X < 748)
           4       316   286  pass (pass if 223 < X < 402)
           5       165   151  pass (pass if 90 < X < 223)
           6+      154   156  pass (pass if 90 < X < 223)
                              PASS (pass if all twelve counts pass)
     No long run (LEN => 34)  PASS (pass if no long run)

BLOCK 2
     Monobit X= 9908          PASS (pass if 9654 < X < 10346)
     Poker X= 19.9808         PASS (pass if 1.03 < X < 57.4)
     Run length  Zeros  Ones
           1      2478  2570  pass (pass if 2267 < X < 2733)
           2      1262  1289  pass (pass if 1079 < X < 1421)
           3       690   559  pass (pass if 502 < X < 748)
           4       298   299  pass (pass if 223 < X < 402)
           5       154   154  pass (pass if 90 < X < 223)
           6+      152   163  pass (pass if 90 < X < 223)
                              PASS (pass if all twelve counts pass)
     No long run (LEN => 34)  PASS (pass if no long run)

BLOCK 3
     Monobit X= 10003         PASS (pass if 9654 < X < 10346)
     Poker X= 15.9232         PASS (pass if 1.03 < X < 57.4)
     Run length  Zeros  Ones
           1      2464  2497  pass (pass if 2267 < X < 2733)
           2      1271  1235  pass (pass if 1079 < X < 1421)
           3       635   647  pass (pass if 502 < X < 748)
           4       313   289  pass (pass if 223 < X < 402)
           5       145   169  pass (pass if 90 < X < 223)
           6+      159   151  pass (pass if 90 < X < 223)
                              PASS (pass if all twelve counts pass)
     No long run (LEN => 34)  PASS (pass if no long run)

BLOCK 4
     Monobit X= 9924          PASS (pass if 9654 < X < 10346)
     Poker X= 8.3264          PASS (pass if 1.03 < X < 57.4)
     Run length  Zeros  Ones
           1      2497  2556  pass (pass if 2267 < X < 2733)
           2      1252  1201  pass (pass if 1079 < X < 1421)
           3       601   634  pass (pass if 502 < X < 748)
           4       319   295  pass (pass if 223 < X < 402)
           5       163   161  pass (pass if 90 < X < 223)
           6+      168   152  pass (pass if 90 < X < 223)
                              PASS (pass if all twelve counts pass)
     No long run (LEN => 34)  PASS (pass if no long run)

BLOCK 5
     Monobit X= 9989          PASS (pass if 9654 < X < 10346)
     Poker X= 21.5488         PASS (pass if 1.03 < X < 57.4)
```

```
     Run length  Zeros  Ones
           1      2458  2491  pass (pass if 2267 < X < 2733)
           2      1228  1220  pass (pass if 1079 < X < 1421)
           3       634   627  pass (pass if 502 < X < 748)
           4       341   311  pass (pass if 223 < X < 402)
           5       156   150  pass (pass if 90 < X < 223)
           6+      151   168  pass (pass if 90 < X < 223)
                              PASS (pass if all twelve counts pass)
     No long run (LEN => 34)  PASS (pass if no long run)
```

```
Results of FIPS 140-1 Specified Tests on block1.rng

BLOCK 1
     Monobit X= 10046         PASS (pass if 9654 < X < 10346)
     Poker X= 8.8384          PASS (pass if 1.03 < X < 57.4)
     Run length  Zeros  Ones
           1      2515  2473  pass (pass if 2267 < X < 2733)
           2      1222  1270  pass (pass if 1079 < X < 1421)
           3       639   594  pass (pass if 502 < X < 748)
           4       318   318  pass (pass if 223 < X < 402)
           5       140   173  pass (pass if 90 < X < 223)
           6+      153   159  pass (pass if 90 < X < 223)
                              PASS (pass if all twelve counts pass)
     No long run (LEN => 34)  PASS (pass if no long run)

BLOCK 2
     Monobit X= 9966          PASS (pass if 9654 < X < 10346)
     Poker X= 15.2576         PASS (pass if 1.03 < X < 57.4)
     Run length  Zeros  Ones
           1      2455  2447  pass (pass if 2267 < X < 2733)
           2      1189  1192  pass (pass if 1079 < X < 1421)
           3       603   642  pass (pass if 502 < X < 748)
           4       343   309  pass (pass if 223 < X < 402)
           5       158   164  pass (pass if 90 < X < 223)
           6+      174   167  pass (pass if 90 < X < 223)
                              PASS (pass if all twelve counts pass)
     No long run (LEN => 34)  PASS (pass if no long run)

BLOCK 3
     Monobit X= 9950          PASS (pass if 9654 < X < 10346)
     Poker X= 19.0464         PASS (pass if 1.03 < X < 57.4)
     Run length  Zeros  Ones
           1      2516  2551  pass (pass if 2267 < X < 2733)
           2      1265  1230  pass (pass if 1079 < X < 1421)
           3       625   638  pass (pass if 502 < X < 748)
           4       271   297  pass (pass if 223 < X < 402)
           5       165   151  pass (pass if 90 < X < 223)
           6+      178   153  pass (pass if 90 < X < 223)
                              PASS (pass if all twelve counts pass)
     No long run (LEN => 34)  PASS (pass if no long run)

BLOCK 4
     Monobit X= 9923          PASS (pass if 9654 < X < 10346)
     Poker X= 18.4896         PASS (pass if 1.03 < X < 57.4)
```

```
     Run length  Zeros  Ones
           1      2431  2496  pass (pass if 2267 < X < 2733)
           2      1289  1232  pass (pass if 1079 < X < 1421)
           3       608   645  pass (pass if 502 < X < 748)
           4       307   291  pass (pass if 223 < X < 402)
           5       173   148  pass (pass if 90 < X < 223)
           6+      161   157  pass (pass if 90 < X < 223)
                              PASS (pass if all twelve counts pass)
     No long run (LEN => 34)  PASS (pass if no long run)

BLOCK 5
     Monobit X= 10119         PASS (pass if 9654 < X < 10346)
     Poker X= 19.168          PASS (pass if 1.03 < X < 57.4)
     Run length  Zeros  Ones
           1      2525  2465  pass (pass if 2267 < X < 2733)
           2      1254  1274  pass (pass if 1079 < X < 1421)
           3       611   597  pass (pass if 502 < X < 748)
           4       302   331  pass (pass if 223 < X < 402)
           5       160   172  pass (pass if 90 < X < 223)
           6+      143   156  pass (pass if 90 < X < 223)
                              PASS (pass if all twelve counts pass)
     No long run (LEN => 34)  PASS (pass if no long run)



Results of FIPS 140-1 Specified Tests on block2.rng

BLOCK 1
     Monobit X= 9980          PASS (pass if 9654 < X < 10346)
     Poker X= 12.1472         PASS (pass if 1.03 < X < 57.4)
     Run length  Zeros  Ones
           1      2461  2468  pass (pass if 2267 < X < 2733)
           2      1234  1230  pass (pass if 1079 < X < 1421)
           3       617   661  pass (pass if 502 < X < 748)
           4       326   290  pass (pass if 223 < X < 402)
           5       140   135  pass (pass if 90 < X < 223)
           6+      180   175  pass (pass if 90 < X < 223)
                              PASS (pass if all twelve counts pass)
     No long run (LEN => 34)  PASS (pass if no long run)

BLOCK 2
     Monobit X= 9941          PASS (pass if 9654 < X < 10346)
     Poker X= 13.472          PASS (pass if 1.03 < X < 57.4)
     Run length  Zeros  Ones
           1      2547  2551  pass (pass if 2267 < X < 2733)
           2      1178  1199  pass (pass if 1079 < X < 1421)
           3       651   638  pass (pass if 502 < X < 748)
           4       315   335  pass (pass if 223 < X < 402)
           5       154   148  pass (pass if 90 < X < 223)
           6+      171   145  pass (pass if 90 < X < 223)
                              PASS (pass if all twelve counts pass)
     No long run (LEN => 34)  PASS (pass if no long run)

BLOCK 3
     Monobit X= 9941          PASS (pass if 9654 < X < 10346)
     Poker X= 19.8336         PASS (pass if 1.03 < X < 57.4)
```

```
     Run length  Zeros  Ones
           1      2539  2559  pass (pass if 2267 < X < 2733)
           2      1202  1203  pass (pass if 1079 < X < 1421)
           3       599   606  pass (pass if 502 < X < 748)
           4       344   320  pass (pass if 223 < X < 402)
           5       163   164  pass (pass if 90 < X < 223)
           6+      159   154  pass (pass if 90 < X < 223)
                              PASS (pass if all twelve counts pass)
     No long run (LEN => 34)  PASS (pass if no long run)

BLOCK 4
     Monobit X= 9974          PASS (pass if 9654 < X < 10346)
     Poker X= 8.9472          PASS (pass if 1.03 < X < 57.4)
     Run length  Zeros  Ones
           1      2523  2542  pass (pass if 2267 < X < 2733)
           2      1312  1307  pass (pass if 1079 < X < 1421)
           3       588   608  pass (pass if 502 < X < 748)
           4       334   305  pass (pass if 223 < X < 402)
           5       151   142  pass (pass if 90 < X < 223)
           6+      151   156  pass (pass if 90 < X < 223)
                              PASS (pass if all twelve counts pass)
     No long run (LEN => 34)  PASS (pass if no long run)

BLOCK 5
     Monobit X= 9998          PASS (pass if 9654 < X < 10346)
     Poker X= 6.7456          PASS (pass if 1.03 < X < 57.4)
     Run length  Zeros  Ones
           1      2477  2463  pass (pass if 2267 < X < 2733)
           2      1249  1257  pass (pass if 1079 < X < 1421)
           3       600   631  pass (pass if 502 < X < 748)
           4       335   319  pass (pass if 223 < X < 402)
           5       175   152  pass (pass if 90 < X < 223)
           6+      143   158  pass (pass if 90 < X < 223)
                              PASS (pass if all twelve counts pass)
     No long run (LEN => 34)  PASS (pass if no long run)



Results of FIPS 140-1 Specified Tests on block3.rng

BLOCK 1
     Monobit X= 10040         PASS (pass if 9654 < X < 10346)
     Poker X= 9.7408          PASS (pass if 1.03 < X < 57.4)
     Run length  Zeros  Ones
           1      2487  2519  pass (pass if 2267 < X < 2733)
           2      1265  1227  pass (pass if 1079 < X < 1421)
           3       659   613  pass (pass if 502 < X < 748)
           4       288   319  pass (pass if 223 < X < 402)
           5       145   159  pass (pass if 90 < X < 223)
           6+      159   166  pass (pass if 90 < X < 223)
                              PASS (pass if all twelve counts pass)
     No long run (LEN => 34)  PASS (pass if no long run)

BLOCK 2
     Monobit X= 10017         PASS (pass if 9654 < X < 10346)
     Poker X= 15.6032         PASS (pass if 1.03 < X < 57.4)
```

```
     Run length  Zeros  Ones
           1      2484  2478  pass (pass if 2267 < X < 2733)
           2      1249  1224  pass (pass if 1079 < X < 1421)
           3       621   647  pass (pass if 502 < X < 748)
           4       278   281  pass (pass if 223 < X < 402)
           5       165   173  pass (pass if 90 < X < 223)
           6+      170   165  pass (pass if 90 < X < 223)
                              PASS (pass if all twelve counts pass)
     No long run (LEN => 34)  PASS (pass if no long run)

BLOCK 3
     Monobit X= 10091         PASS (pass if 9654 < X < 10346)
     Poker X= 7.9232          PASS (pass if 1.03 < X < 57.4)
     Run length  Zeros  Ones
           1      2560  2473  pass (pass if 2267 < X < 2733)
           2      1220  1265  pass (pass if 1079 < X < 1421)
           3       624   654  pass (pass if 502 < X < 748)
           4       320   312  pass (pass if 223 < X < 402)
           5       133   152  pass (pass if 90 < X < 223)
           6+      161   161  pass (pass if 90 < X < 223)
                              PASS (pass if all twelve counts pass)
     No long run (LEN => 34)  PASS (pass if no long run)

BLOCK 4
     Monobit X= 9992          PASS (pass if 9654 < X < 10346)
     Poker X= 9.8752          PASS (pass if 1.03 < X < 57.4)
     Run length  Zeros  Ones
           1      2463  2444  pass (pass if 2267 < X < 2733)
           2      1243  1286  pass (pass if 1079 < X < 1421)
           3       644   603  pass (pass if 502 < X < 748)
           4       298   332  pass (pass if 223 < X < 402)
           5       158   151  pass (pass if 90 < X < 223)
           6+      164   154  pass (pass if 90 < X < 223)
                              PASS (pass if all twelve counts pass)
     No long run (LEN => 34)  PASS (pass if no long run)

BLOCK 5
     Monobit X= 10023         PASS (pass if 9654 < X < 10346)
     Poker X= 11.04           PASS (pass if 1.03 < X < 57.4)
     Run length  Zeros  Ones
           1      2526  2475  pass (pass if 2267 < X < 2733)
           2      1272  1274  pass (pass if 1079 < X < 1421)
           3       608   639  pass (pass if 502 < X < 748)
           4       302   333  pass (pass if 223 < X < 402)
           5       139   161  pass (pass if 90 < X < 223)
           6+      171   135  pass (pass if 90 < X < 223)
                              PASS (pass if all twelve counts pass)
     No long run (LEN => 34)  PASS (pass if no long run)



Results of FIPS 140-1 Specified Tests on block4.rng

BLOCK 1
     Monobit X= 9993          PASS (pass if 9654 < X < 10346)
     Poker X= 17.7152         PASS (pass if 1.03 < X < 57.4)
```

```
     Run length  Zeros  Ones
           1      2568  2519  pass (pass if 2267 < X < 2733)
           2      1249  1280  pass (pass if 1079 < X < 1421)
           3       600   627  pass (pass if 502 < X < 748)
           4       314   302  pass (pass if 223 < X < 402)
           5       142   150  pass (pass if 90 < X < 223)
           6+      163   158  pass (pass if 90 < X < 223)
                              PASS (pass if all twelve counts pass)
     No long run (LEN => 34)  PASS (pass if no long run)

BLOCK 2
     Monobit X= 9947          PASS (pass if 9654 < X < 10346)
     Poker X= 4.288           PASS (pass if 1.03 < X < 57.4)
     Run length  Zeros  Ones
           1      2503  2492  pass (pass if 2267 < X < 2733)
           2      1207  1211  pass (pass if 1079 < X < 1421)
           3       582   645  pass (pass if 502 < X < 748)
           4       335   309  pass (pass if 223 < X < 402)
           5       171   155  pass (pass if 90 < X < 223)
           6+      172   158  pass (pass if 90 < X < 223)
                              PASS (pass if all twelve counts pass)
     No long run (LEN => 34)  PASS (pass if no long run)

BLOCK 3
     Monobit X= 10000         PASS (pass if 9654 < X < 10346)
     Poker X= 19.68           PASS (pass if 1.03 < X < 57.4)
     Run length  Zeros  Ones
           1      2453  2517  pass (pass if 2267 < X < 2733)
           2      1299  1212  pass (pass if 1079 < X < 1421)
           3       636   626  pass (pass if 502 < X < 748)
           4       311   328  pass (pass if 223 < X < 402)
           5       156   157  pass (pass if 90 < X < 223)
           6+      146   160  pass (pass if 90 < X < 223)
                              PASS (pass if all twelve counts pass)
     No long run (LEN => 34)  PASS (pass if no long run)

BLOCK 4
     Monobit X= 9984          PASS (pass if 9654 < X < 10346)
     Poker X= 14.3936         PASS (pass if 1.03 < X < 57.4)
     Run length  Zeros  Ones
           1      2465  2495  pass (pass if 2267 < X < 2733)
           2      1315  1231  pass (pass if 1079 < X < 1421)
           3       594   665  pass (pass if 502 < X < 748)
           4       321   296  pass (pass if 223 < X < 402)
           5       154   164  pass (pass if 90 < X < 223)
           6+      150   148  pass (pass if 90 < X < 223)
                              PASS (pass if all twelve counts pass)
     No long run (LEN => 34)  PASS (pass if no long run)

BLOCK 5
     Monobit X= 9974          PASS (pass if 9654 < X < 10346)
     Poker X= 11.4752         PASS (pass if 1.03 < X < 57.4)
     Run length  Zeros  Ones
           1      2483  2452  pass (pass if 2267 < X < 2733)
           2      1261  1333  pass (pass if 1079 < X < 1421)
           3       610   576  pass (pass if 502 < X < 748)
           4       330   340  pass (pass if 223 < X < 402)
```

```
            5       149   145  pass (pass if 90 < X < 223)
            6+      160   147  pass (pass if 90 < X < 223)
                               PASS (pass if all twelve counts pass)
      No long run (LEN => 34)  PASS (pass if no long run)
```

Results of FIPS 140-1 Specified Tests on block5.rng

```
BLOCK 1
      Monobit X= 9970          PASS (pass if 9654 < X < 10346)
      Poker X= 31.3792         PASS (pass if 1.03 < X < 57.4)
      Run length  Zeros  Ones
            1      2501  2522  pass (pass if 2267 < X < 2733)
            2      1256  1257  pass (pass if 1079 < X < 1421)
            3       635   607  pass (pass if 502 < X < 748)
            4       311   324  pass (pass if 223 < X < 402)
            5       169   168  pass (pass if 90 < X < 223)
            6+      143   136  pass (pass if 90 < X < 223)
                               PASS (pass if all twelve counts pass)
      No long run (LEN => 34)  PASS (pass if no long run)

BLOCK 2
      Monobit X= 9915          PASS (pass if 9654 < X < 10346)
      Poker X= 16.6656         PASS (pass if 1.03 < X < 57.4)
      Run length  Zeros  Ones
            1      2520  2575  pass (pass if 2267 < X < 2733)
            2      1253  1198  pass (pass if 1079 < X < 1421)
            3       608   629  pass (pass if 502 < X < 748)
            4       302   327  pass (pass if 223 < X < 402)
            5       153   138  pass (pass if 90 < X < 223)
            6+      182   152  pass (pass if 90 < X < 223)
                               PASS (pass if all twelve counts pass)
      No long run (LEN => 34)  PASS (pass if no long run)

BLOCK 3
      Monobit X= 10034         PASS (pass if 9654 < X < 10346)
      Poker X= 14.0096         PASS (pass if 1.03 < X < 57.4)
      Run length  Zeros  Ones
            1      2518  2491  pass (pass if 2267 < X < 2733)
            2      1274  1257  pass (pass if 1079 < X < 1421)
            3       587   620  pass (pass if 502 < X < 748)
            4       306   322  pass (pass if 223 < X < 402)
            5       165   171  pass (pass if 90 < X < 223)
            6+      158   146  pass (pass if 90 < X < 223)
                               PASS (pass if all twelve counts pass)
      No long run (LEN => 34)  PASS (pass if no long run)

BLOCK 4
      Monobit X= 9997          PASS (pass if 9654 < X < 10346)
      Poker X= 18.5984         PASS (pass if 1.03 < X < 57.4)
      Run length  Zeros  Ones
            1      2494  2414  pass (pass if 2267 < X < 2733)
            2      1206  1294  pass (pass if 1079 < X < 1421)
            3       580   588  pass (pass if 502 < X < 748)
            4       339   346  pass (pass if 223 < X < 402)
```

```
            5       171   163  pass (pass if 90 < X < 223)
            6+      165   150  pass (pass if 90 < X < 223)
                               PASS (pass if all twelve counts pass)
      No long run (LEN => 34)  PASS (pass if no long run)

 BLOCK 5
      Monobit X= 9977          PASS (pass if 9654 < X < 10346)
      Poker X= 34.2144         PASS (pass if 1.03 < X < 57.4)
      Run length  Zeros  Ones
            1      2342  2423  pass (pass if 2267 < X < 2733)
            2      1279  1239  pass (pass if 1079 < X < 1421)
            3       647   627  pass (pass if 502 < X < 748)
            4       342   303  pass (pass if 223 < X < 402)
            5       152   165  pass (pass if 90 < X < 223)
            6+      150   156  pass (pass if 90 < X < 223)
                               PASS (pass if all twelve counts pass)
      No long run (LEN => 34)  PASS (pass if no long run)



 Results of FIPS 140-1 Specified Tests on block6.rng

 BLOCK 1
      Monobit X= 10008         PASS (pass if 9654 < X < 10346)
      Poker X= 16.4928         PASS (pass if 1.03 < X < 57.4)
      Run length  Zeros  Ones
            1      2512  2491  pass (pass if 2267 < X < 2733)
            2      1246  1233  pass (pass if 1079 < X < 1421)
            3       601   624  pass (pass if 502 < X < 748)
            4       300   349  pass (pass if 223 < X < 402)
            5       158   135  pass (pass if 90 < X < 223)
            6+      168   153  pass (pass if 90 < X < 223)
                               PASS (pass if all twelve counts pass)
      No long run (LEN => 34)  PASS (pass if no long run)

 BLOCK 2
      Monobit X= 9970          PASS (pass if 9654 < X < 10346)
      Poker X= 8.1088          PASS (pass if 1.03 < X < 57.4)
      Run length  Zeros  Ones
            1      2540  2521  pass (pass if 2267 < X < 2733)
            2      1213  1289  pass (pass if 1079 < X < 1421)
            3       647   590  pass (pass if 502 < X < 748)
            4       299   285  pass (pass if 223 < X < 402)
            5       170   170  pass (pass if 90 < X < 223)
            6+      150   164  pass (pass if 90 < X < 223)
                               PASS (pass if all twelve counts pass)
      No long run (LEN => 34)  PASS (pass if no long run)

 BLOCK 3
      Monobit X= 10022         PASS (pass if 9654 < X < 10346)
      Poker X= 14.7072         PASS (pass if 1.03 < X < 57.4)
      Run length  Zeros  Ones
            1      2547  2521  pass (pass if 2267 < X < 2733)
            2      1241  1256  pass (pass if 1079 < X < 1421)
            3       623   610  pass (pass if 502 < X < 748)
            4       315   346  pass (pass if 223 < X < 402)
```

```
          5       138   139  pass (pass if 90 < X < 223)
          6+      165   158  pass (pass if 90 < X < 223)
                             PASS (pass if all twelve counts pass)
     No long run (LEN => 34)  PASS (pass if no long run)

BLOCK 4
     Monobit X= 9911          PASS (pass if 9654 < X < 10346)
     Poker X= 17.9264         PASS (pass if 1.03 < X < 57.4)
     Run length  Zeros  Ones
          1      2490  2498  pass (pass if 2267 < X < 2733)
          2      1213  1173  pass (pass if 1079 < X < 1421)
          3       614   692  pass (pass if 502 < X < 748)
          4       286   302  pass (pass if 223 < X < 402)
          5       181   158  pass (pass if 90 < X < 223)
          6+      182   143  pass (pass if 90 < X < 223)
                             PASS (pass if all twelve counts pass)
     No long run (LEN => 34)  PASS (pass if no long run)

BLOCK 5
     Monobit X= 10036         PASS (pass if 9654 < X < 10346)
     Poker X= 14.0736         PASS (pass if 1.03 < X < 57.4)
     Run length  Zeros  Ones
          1      2534  2471  pass (pass if 2267 < X < 2733)
          2      1252  1286  pass (pass if 1079 < X < 1421)
          3       619   645  pass (pass if 502 < X < 748)
          4       320   353  pass (pass if 223 < X < 402)
          5       164   142  pass (pass if 90 < X < 223)
          6+      141   133  pass (pass if 90 < X < 223)
                             PASS (pass if all twelve counts pass)
     No long run (LEN => 34)  PASS (pass if no long run)



Results of FIPS 140-1 Specified Tests on block7.rng

BLOCK 1
     Monobit X= 10056         PASS (pass if 9654 < X < 10346)
     Poker X= 12.2112         PASS (pass if 1.03 < X < 57.4)
     Run length  Zeros  Ones
          1      2572  2500  pass (pass if 2267 < X < 2733)
          2      1199  1304  pass (pass if 1079 < X < 1421)
          3       640   602  pass (pass if 502 < X < 748)
          4       322   311  pass (pass if 223 < X < 402)
          5       140   140  pass (pass if 90 < X < 223)
          6+      153   168  pass (pass if 90 < X < 223)
                             PASS (pass if all twelve counts pass)
     No long run (LEN => 34)  PASS (pass if no long run)

BLOCK 2
     Monobit X= 9980          PASS (pass if 9654 < X < 10346)
     Poker X= 21.9456         PASS (pass if 1.03 < X < 57.4)
     Run length  Zeros  Ones
          1      2450  2489  pass (pass if 2267 < X < 2733)
          2      1270  1233  pass (pass if 1079 < X < 1421)
          3       643   632  pass (pass if 502 < X < 748)
          4       324   314  pass (pass if 223 < X < 402)
```

```
        5       135   172  pass (pass if 90 < X < 223)
        6+      162   144  pass (pass if 90 < X < 223)
                           PASS (pass if all twelve counts pass)
  No long run (LEN => 34)  PASS (pass if no long run)

BLOCK 3
  Monobit X= 9986          PASS (pass if 9654 < X < 10346)
  Poker X= 24.5824         PASS (pass if 1.03 < X < 57.4)
  Run length  Zeros  Ones
        1      2511  2525  pass (pass if 2267 < X < 2733)
        2      1289  1278  pass (pass if 1079 < X < 1421)
        3       641   640  pass (pass if 502 < X < 748)
        4       308   313  pass (pass if 223 < X < 402)
        5       152   154  pass (pass if 90 < X < 223)
        6+      147   138  pass (pass if 90 < X < 223)
                           PASS (pass if all twelve counts pass)
  No long run (LEN => 34)  PASS (pass if no long run)

BLOCK 4
  Monobit X= 10035         PASS (pass if 9654 < X < 10346)
  Poker X= 18.4704         PASS (pass if 1.03 < X < 57.4)
  Run length  Zeros  Ones
        1      2576  2538  pass (pass if 2267 < X < 2733)
        2      1251  1220  pass (pass if 1079 < X < 1421)
        3       606   648  pass (pass if 502 < X < 748)
        4       316   337  pass (pass if 223 < X < 402)
        5       148   173  pass (pass if 90 < X < 223)
        6+      150   131  pass (pass if 90 < X < 223)
                           PASS (pass if all twelve counts pass)
  No long run (LEN => 34)  PASS (pass if no long run)

BLOCK 5
  Monobit X= 10029         PASS (pass if 9654 < X < 10346)
  Poker X= 9.1776          PASS (pass if 1.03 < X < 57.4)
  Run length  Zeros  Ones
        1      2438  2472  pass (pass if 2267 < X < 2733)
        2      1274  1215  pass (pass if 1079 < X < 1421)
        3       641   643  pass (pass if 502 < X < 748)
        4       310   322  pass (pass if 223 < X < 402)
        5       167   154  pass (pass if 90 < X < 223)
        6+      140   164  pass (pass if 90 < X < 223)
                           PASS (pass if all twelve counts pass)
  No long run (LEN => 34)  PASS (pass if no long run)
```

Appendix B: Diehard Test Result Details

```
       NOTE: Most of the tests in DIEHARD return a p-value, which
       should be uniform on [0,1) if the input file contains truly
       independent random bits.   Those p-values are obtained by
       p=F(X), where F is the assumed distribution of the sample
       random variable X---often normal. But that assumed F is just
       an asymptotic approximation, for which the fit will be worst
       in the tails. Thus you should not be surprised with
       occasional p-values near 0 or 1, such as .0012 or .9983.
       When a bit stream really FAILS BIG, you will get p's of 0 or
       1 to six or more places.  By all means, do not, as a
       Statistician might, think that a p < .025 or p> .975 means
       that the RNG has "failed the test at the .05 level".  Such
       p's happen among the hundreds that DIEHARD produces, even
       with good RNG's.  So keep in mind that " p happens".
     :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
     ::            This is the BIRTHDAY SPACINGS TEST                 ::
     :: Choose m birthdays in a year of n days.  List the spacings    ::
     :: between the birthdays.  If j is the number of values that     ::
     :: occur more than once in that list, then j is asymptotically   ::
     :: Poisson distributed with mean m^3/(4n).  Experience shows n   ::
     :: must be quite large, say n>=2^18, for comparing the results   ::
     :: to the Poisson distribution with that mean.  This test uses   ::
     :: n=2^24 and m=2^9,  so that the underlying distribution for j  ::
     :: is taken to be Poisson with lambda=2^27/(2^26)=2.  A sample   ::
     :: of 500 j's is taken, and a chi-square goodness of fit test    ::
     :: provides a p value.  The first test uses bits 1-24 (counting  ::
     :: from the left) from integers in the specified file.           ::
     ::   Then the file is closed and reopened. Next, bits 2-25 are   ::
     :: used to provide birthdays, then 3-26 and so on to bits 9-32.  ::
     :: Each set of bits provides a p-value, and the nine p-values    ::
     :: provide a sample for a KSTEST.                                ::
     :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
 BIRTHDAY SPACINGS TEST, M= 512 N=2**24 LAMBDA=  2.0000
           Results for block0.rng
                   For a sample of size 500:     mean
           block0.rng      using bits  1 to 24   1.916
  duplicate       number       number
  spacings       observed     expected
        0          61.       67.668
        1         141.      135.335
        2         157.      135.335
        3          88.       90.224
        4          35.       45.112
        5          11.       18.045
  6 to INF          7.        8.282
 Chisquare with  6 d.o.f. =     9.63 p-value=  .858982
  :::::::::::::::::::::::::::::::::::::::::
                   For a sample of size 500:     mean
           block0.rng      using bits  2 to 25   1.910
  duplicate       number       number
  spacings       observed     expected
        0          69.       67.668
        1         142.      135.335
        2         144.      135.335
```

```
      3          85.       90.224
      4          44.       45.112
      5           7.       18.045
 6 to INF         9.        8.282
Chisquare with  6 d.o.f. =     8.06 p-value=  .766366
 :::::::::::::::::::::::::::::::::::::::::
                 For a sample of size 500:     mean
          block0.rng      using bits  3 to 26   1.996
 duplicate       number       number
 spacings       observed     expected
      0          65.       67.668
      1         145.      135.335
      2         130.      135.335
      3          89.       90.224
      4          47.       45.112
      5          14.       18.045
 6 to INF        10.        8.282
Chisquare with  6 d.o.f. =     2.36 p-value=  .116675
 :::::::::::::::::::::::::::::::::::::::::
                 For a sample of size 500:     mean
          block0.rng      using bits  4 to 27   2.022
 duplicate       number       number
 spacings       observed     expected
      0          72.       67.668
      1         132.      135.335
      2         118.      135.335
      3          97.       90.224
      4          55.       45.112
      5          24.       18.045
 6 to INF         2.        8.282
Chisquare with  6 d.o.f. =    11.99 p-value=  .937734
 :::::::::::::::::::::::::::::::::::::::::
                 For a sample of size 500:     mean
          block0.rng      using bits  5 to 28   1.960
 duplicate       number       number
 spacings       observed     expected
      0          69.       67.668
      1         140.      135.335
      2         136.      135.335
      3          83.       90.224
      4          49.       45.112
      5          15.       18.045
 6 to INF         8.        8.282
Chisquare with  6 d.o.f. =     1.63 p-value=  .049389
 :::::::::::::::::::::::::::::::::::::::::
                 For a sample of size 500:     mean
          block0.rng      using bits  6 to 29   2.024
 duplicate       number       number
 spacings       observed     expected
      0          54.       67.668
      1         138.      135.335
      2         147.      135.335
      3          89.       90.224
      4          53.       45.112
      5          16.       18.045
 6 to INF         3.        8.282
Chisquare with  6 d.o.f. =     8.81 p-value=  .815725
```

```
 :::::::::::::::::::::::::::::::::::::::::
                  For a sample of size 500:     mean
          block0.rng      using bits  7 to 30   2.038
 duplicate       number       number
 spacings       observed     expected
       0          72.       67.668
       1         114.      135.335
       2         144.      135.335
       3          97.       90.224
       4          48.       45.112
       5          16.       18.045
 6 to INF          9.        8.282
Chisquare with  6 d.o.f. =     5.18 p-value=  .479490
 :::::::::::::::::::::::::::::::::::::::::
                  For a sample of size 500:     mean
          block0.rng      using bits  8 to 31   1.972
 duplicate       number       number
 spacings       observed     expected
       0          68.       67.668
       1         151.      135.335
       2         125.      135.335
       3          81.       90.224
       4          49.       45.112
       5          15.       18.045
 6 to INF         11.        8.282
Chisquare with  6 d.o.f. =     5.29 p-value=  .492559
 :::::::::::::::::::::::::::::::::::::::::
                  For a sample of size 500:     mean
          block0.rng      using bits  9 to 32   2.026
 duplicate       number       number
 spacings       observed     expected
       0          63.       67.668
       1         141.      135.335
       2         133.      135.335
       3          82.       90.224
       4          53.       45.112
       5          21.       18.045
 6 to INF          7.        8.282
Chisquare with  6 d.o.f. =     3.41 p-value=  .244183
 :::::::::::::::::::::::::::::::::::::::::
  The 9 p-values were
       .858982   .766366   .116675   .937734   .049389
       .815725   .479490   .492559   .244183
 A KSTEST for the 9 p-values yields  .079935

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

    ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
    ::             THE OVERLAPPING 5-PERMUTATION TEST                 ::
    :: This is the OPERM5 test.  It looks at a sequence of one mill- ::
    :: ion 32-bit random integers.  Each set of five consecutive     ::
    :: integers can be in one of 120 states, for the 5! possible or- ::
    :: derings of five numbers.  Thus the 5th, 6th, 7th,...numbers   ::
    :: each provide a state. As many thousands of state transitions  ::
    :: are observed,  cumulative counts are made of the number of    ::
    :: occurences of each state.  Then the quadratic form in the     ::
    :: weak inverse of the 120x120 covariance matrix yields a test   ::
```

```
     :: equivalent to the likelihood ratio test that the 120 cell     ::
     :: counts came from the specified (asymptotically) normal dis-   ::
     :: tribution with the specified 120x120 covariance matrix (with  ::
     :: rank 99).  This version uses 1,000,000 integers, twice.       ::
     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
           OPERM5 test for file block0.rng
     For a sample of 1,000,000 consecutive 5-tuples,
 chisquare for 99 degrees of freedom= 71.452; p-value= .016727
           OPERM5 test for file block0.rng
     For a sample of 1,000,000 consecutive 5-tuples,
 chisquare for 99 degrees of freedom=107.158; p-value= .729698
     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
     :: This is the BINARY RANK TEST for 31x31 matrices. The leftmost ::
     :: 31 bits of 31 random integers from the test sequence are used ::
     :: to form a 31x31 binary matrix over the field {0,1}. The rank  ::
     :: is determined. That rank can be from 0 to 31, but ranks< 28   ::
     :: are rare, and their counts are pooled with those for rank 28. ::
     :: Ranks are found for 40,000 such random matrices and a chisqua-::
     :: re test is performed on counts for ranks 31,30,29 and <=28.   ::
     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
    Binary rank test for block0.rng
         Rank test for 31x31 binary matrices:
        rows from leftmost 31 bits of each 32-bit integer
      rank   observed  expected (o-e)^2/e  sum
        28       230     211.4  1.633211    1.633
        29      5098    5134.0   .252578    1.886
        30     23055   23103.0   .099922    1.986
        31     11617   11551.5   .371124    2.357
  chisquare= 2.357 for 3 d. of f.; p-value= .560648
--------------------------------------------------------------
     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
     :: This is the BINARY RANK TEST for 32x32 matrices. A random 32x ::
     :: 32 binary matrix is formed, each row a 32-bit random integer. ::
     :: The rank is determined. That rank can be from 0 to 32, ranks  ::
     :: less than 29 are rare, and their counts are pooled with those ::
     :: for rank 29.  Ranks are found for 40,000 such random matrices ::
     :: and a chisquare test is performed on counts for ranks  32,31, ::
     :: 30 and <=29.                                                  ::
     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
    Binary rank test for block0.rng
         Rank test for 32x32 binary matrices:
        rows from leftmost 32 bits of each 32-bit integer
      rank   observed  expected (o-e)^2/e  sum
        29       179     211.4  4.970852    4.971
        30      5046    5134.0  1.508724    6.480
        31     23244   23103.0   .859964    7.340
        32     11531   11551.5   .036467    7.376
  chisquare= 7.376 for 3 d. of f.; p-value= .942474
--------------------------------------------------------------

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
     :: This is the BINARY RANK TEST for 6x8 matrices.  From each of  ::
     :: six random 32-bit integers from the generator under test, a   ::
     :: specified byte is chosen, and the resulting six bytes form a  ::
     :: 6x8 binary matrix whose rank is determined.  That rank can be ::
```

```
:: from 0 to 6, but ranks 0,1,2,3 are rare; their counts are     ::
:: pooled with those for rank 4. Ranks are found for 100,000     ::
:: random matrices, and a chi-square test is performed on        ::
:: counts for ranks 6,5 and <=4.                                 ::
:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
    Binary Rank Test for block0.rng
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block0.rng
b-rank test for bits  1 to  8
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          983       944.3       1.586       1.586
     r =5        21743     21743.9        .000       1.586
     r =6        77274     77311.8        .018       1.604
                   p=1-exp(-SUM/2)= .55167
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block0.rng
b-rank test for bits  2 to  9
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          947       944.3        .008        .008
     r =5        21737     21743.9        .002        .010
     r =6        77316     77311.8        .000        .010
                   p=1-exp(-SUM/2)= .00505
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block0.rng
b-rank test for bits  3 to 10
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          937       944.3        .056        .056
     r =5        21629     21743.9        .607        .664
     r =6        77434     77311.8        .193        .857
                   p=1-exp(-SUM/2)= .34843
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block0.rng
b-rank test for bits  4 to 11
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          939       944.3        .030        .030
     r =5        21771     21743.9        .034        .064
     r =6        77290     77311.8        .006        .070
                   p=1-exp(-SUM/2)= .03424
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block0.rng
b-rank test for bits  5 to 12
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          947       944.3        .008        .008
     r =5        21644     21743.9        .459        .467
     r =6        77409     77311.8        .122        .589
                   p=1-exp(-SUM/2)= .25505
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block0.rng
b-rank test for bits  6 to 13
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          935       944.3        .092        .092
     r =5        21715     21743.9        .038        .130
     r =6        77350     77311.8        .019        .149
                   p=1-exp(-SUM/2)= .07175
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block0.rng
b-rank test for bits  7 to 14
```

```
               OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4         943       944.3        .002        .002
     r =5       21622     21743.9        .683        .685
     r =6       77435     77311.8        .196        .881
                  p=1-exp(-SUM/2)= .35645
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block0.rng
b-rank test for bits  8 to 15
               OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4         954       944.3        .100        .100
     r =5       21723     21743.9        .020        .120
     r =6       77323     77311.8        .002        .121
                  p=1-exp(-SUM/2)= .05886
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block0.rng
b-rank test for bits  9 to 16
               OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4         945       944.3        .001        .001
     r =5       21999     21743.9       2.993       2.993
     r =6       77056     77311.8        .846       3.840
                  p=1-exp(-SUM/2)= .85337
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block0.rng
b-rank test for bits 10 to 17
               OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4         950       944.3        .034        .034
     r =5       21714     21743.9        .041        .076
     r =6       77336     77311.8        .008        .083
                  p=1-exp(-SUM/2)= .04069
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block0.rng
b-rank test for bits 11 to 18
               OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4         940       944.3        .020        .020
     r =5       21783     21743.9        .070        .090
     r =6       77277     77311.8        .016        .106
                  p=1-exp(-SUM/2)= .05142
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block0.rng
b-rank test for bits 12 to 19
               OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4         967       944.3        .546        .546
     r =5       21873     21743.9        .767       1.312
     r =6       77160     77311.8        .298       1.610
                  p=1-exp(-SUM/2)= .55296
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block0.rng
b-rank test for bits 13 to 20
               OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4         956       944.3        .145        .145
     r =5       21747     21743.9        .000        .145
     r =6       77297     77311.8        .003        .148
                  p=1-exp(-SUM/2)= .07143
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block0.rng
b-rank test for bits 14 to 21
               OBSERVED   EXPECTED     (O-E)^2/E      SUM
```

```
     r<=4          932       944.3         .160         .160
     r =5        21683     21743.9         .171         .331
     r =6        77385     77311.8         .069         .400
                   p=1-exp(-SUM/2)= .18132
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block0.rng
b-rank test for bits 15 to 22
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          926       944.3         .355         .355
     r =5        21735     21743.9         .004         .358
     r =6        77339     77311.8         .010         .368
                   p=1-exp(-SUM/2)= .16802
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block0.rng
b-rank test for bits 16 to 23
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          914       944.3         .972         .972
     r =5        21719     21743.9         .029        1.001
     r =6        77367     77311.8         .039        1.040
                   p=1-exp(-SUM/2)= .40555
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block0.rng
b-rank test for bits 17 to 24
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          943       944.3         .002         .002
     r =5        21508     21743.9        2.559        2.561
     r =6        77549     77311.8         .728        3.289
                   p=1-exp(-SUM/2)= .80687
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block0.rng
b-rank test for bits 18 to 25
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          929       944.3         .248         .248
     r =5        21505     21743.9        2.625        2.873
     r =6        77566     77311.8         .836        3.709
                   p=1-exp(-SUM/2)= .84343
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block0.rng
b-rank test for bits 19 to 26
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          942       944.3         .006         .006
     r =5        21726     21743.9         .015         .020
     r =6        77332     77311.8         .005         .026
                   p=1-exp(-SUM/2)= .01273
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block0.rng
b-rank test for bits 20 to 27
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          924       944.3         .436         .436
     r =5        21811     21743.9         .207         .644
     r =6        77265     77311.8         .028         .672
                   p=1-exp(-SUM/2)= .28532
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block0.rng
b-rank test for bits 21 to 28
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          953       944.3         .080         .080
```

```
        r =5        21648     21743.9        .423        .503
        r =6        77399     77311.8        .098        .601
                     p=1-exp(-SUM/2)= .25971
      Rank of a 6x8 binary matrix,
   rows formed from eight bits of the RNG block0.rng
   b-rank test for bits 22 to 29
                 OBSERVED   EXPECTED     (O-E)^2/E      SUM
        r<=4          904       944.3      1.720       1.720
        r =5        21816     21743.9        .239       1.959
        r =6        77280     77311.8        .013       1.972
                     p=1-exp(-SUM/2)= .62696
      Rank of a 6x8 binary matrix,
   rows formed from eight bits of the RNG block0.rng
   b-rank test for bits 23 to 30
                 OBSERVED   EXPECTED     (O-E)^2/E      SUM
        r<=4          921       944.3        .575        .575
        r =5        21868     21743.9        .708       1.283
        r =6        77211     77311.8        .131       1.415
                     p=1-exp(-SUM/2)= .50705
      Rank of a 6x8 binary matrix,
   rows formed from eight bits of the RNG block0.rng
   b-rank test for bits 24 to 31
                 OBSERVED   EXPECTED     (O-E)^2/E      SUM
        r<=4          969       944.3        .646        .646
        r =5        22030     21743.9       3.764       4.410
        r =6        77001     77311.8       1.249       5.660
                     p=1-exp(-SUM/2)= .94098
      Rank of a 6x8 binary matrix,
   rows formed from eight bits of the RNG block0.rng
   b-rank test for bits 25 to 32
                 OBSERVED   EXPECTED     (O-E)^2/E      SUM
        r<=4          939       944.3        .030        .030
        r =5        21846     21743.9        .479        .509
        r =6        77215     77311.8        .121        .630
                     p=1-exp(-SUM/2)= .27035
  TEST SUMMARY, 25 tests on 100,000 random 6x8 matrices
 These should be 25 uniform [0,1] random variables:
    .551670     .005052     .348434     .034243     .255055
    .071746     .356446     .058859     .853374     .040689
    .051416     .552956     .071426     .181316     .168023
    .405552     .806872     .843431     .012728     .285323
    .259714     .626962     .507048     .940984     .270353
  brank test summary for block0.rng
      The KS test for those 25 supposed UNI's yields
                   KS p-value= .995340

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

    ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
    ::                   THE BITSTREAM TEST                            ::
    :: The file under test is viewed as a stream of bits. Call them  ::
    :: b1,b2,... .  Consider an alphabet with two "letters", 0 and 1 ::
    :: and think of the stream of bits as a succession of 20-letter  ::
    :: "words", overlapping.  Thus the first word is b1b2...b20, the ::
    :: second is b2b3...b21, and so on.  The bitstream test counts   ::
    :: the number of missing 20-letter (20-bit) words in a string of ::
    :: 2^21 overlapping 20-letter words.  There are 2^20 possible 20 ::
```

```
     :: letter words.  For a truly random string of 2^21+19 bits, the ::
     :: number of missing words j should be (very close to) normally  ::
     :: distributed with mean 141,909 and sigma 428.  Thus            ::
     ::  (j-141909)/428 should be a standard normal variate (z score) ::
     :: that leads to a uniform [0,1) p value.  The test is repeated  ::
     :: twenty times.                                                 ::
     :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
 THE OVERLAPPING 20-tuples BITSTREAM  TEST, 20 BITS PER WORD, N words
   This test uses N=2^21 and samples the bitstream 20 times.
  No. missing words should average  141909. with sigma=428.
---------------------------------------------------------
 tst no  1:  141905 missing words,    -.01 sigmas from mean, p-value= .49597
 tst no  2:  141555 missing words,    -.83 sigmas from mean, p-value= .20387
 tst no  3:  142103 missing words,     .45 sigmas from mean, p-value= .67455
 tst no  4:  141847 missing words,    -.15 sigmas from mean, p-value= .44211
 tst no  5:  142169 missing words,     .61 sigmas from mean, p-value= .72798
 tst no  6:  140896 missing words,   -2.37 sigmas from mean, p-value= .00895
 tst no  7:  142178 missing words,     .63 sigmas from mean, p-value= .73491
 tst no  8:  142889 missing words,    2.29 sigmas from mean, p-value= .98896
 tst no  9:  141382 missing words,   -1.23 sigmas from mean, p-value= .10896
 tst no 10:  141974 missing words,     .15 sigmas from mean, p-value= .56005
 tst no 11:  142468 missing words,    1.31 sigmas from mean, p-value= .90411
 tst no 12:  141087 missing words,   -1.92 sigmas from mean, p-value= .02735
 tst no 13:  142862 missing words,    2.23 sigmas from mean, p-value= .98699
 tst no 14:  141593 missing words,    -.74 sigmas from mean, p-value= .22993
 tst no 15:  142527 missing words,    1.44 sigmas from mean, p-value= .92551
 tst no 16:  141849 missing words,    -.14 sigmas from mean, p-value= .44395
 tst no 17:  141384 missing words,   -1.23 sigmas from mean, p-value= .10984
 tst no 18:  142783 missing words,    2.04 sigmas from mean, p-value= .97939
 tst no 19:  141599 missing words,    -.73 sigmas from mean, p-value= .23421
 tst no 20:  141377 missing words,   -1.24 sigmas from mean, p-value= .10679

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

     :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
     ::             The tests OPSO, OQSO and DNA                      ::
     ::         OPSO means Overlapping-Pairs-Sparse-Occupancy         ::
     :: The OPSO test considers 2-letter words from an alphabet of    ::
     :: 1024 letters.  Each letter is determined by a specified ten   ::
     :: bits from a 32-bit integer in the sequence to be tested. OPSO ::
     :: generates  2^21 (overlapping) 2-letter words  (from 2^21+1    ::
     :: "keystrokes")  and counts the number of missing words---that  ::
     :: is 2-letter words which do not appear in the entire sequence. ::
     :: That count should be very close to normally distributed with  ::
     :: mean 141,909, sigma 290. Thus (missingwrds-141909)/290 should ::
     :: be a standard normal variable. The OPSO test takes 32 bits at ::
     :: a time from the test file and uses a designated set of ten    ::
     :: consecutive bits. It then restarts the file for the next de-  ::
     :: signated 10 bits, and so on.                                  ::
     ::                                                               ::
     ::     OQSO means Overlapping-Quadruples-Sparse-Occupancy        ::
     ::   The test OQSO is similar, except that it considers 4-letter ::
     :: words from an alphabet of 32 letters, each letter determined  ::
     :: by a designated string of 5 consecutive bits from the test    ::
     :: file, elements of which are assumed 32-bit random integers.   ::
     :: The mean number of missing words in a sequence of 2^21 four-  ::
     :: letter words,  (2^21+3 "keystrokes"), is again 141909, with   ::
```

```
    :: sigma = 295.  The mean is based on theory; sigma comes from   ::
    :: extensive simulation.                                         ::
    ::                                                               ::
    ::    The DNA test considers an alphabet of 4 letters::  C,G,A,T,::
    :: determined by two designated bits in the sequence of random   ::
    :: integers being tested.  It considers 10-letter words, so that ::
    :: as in OPSO and OQSO, there are 2^20 possible words, and the   ::
    :: mean number of missing words from a string of 2^21  (over-    ::
    :: lapping)  10-letter  words (2^21+9 "keystrokes") is 141909.   ::
    :: The standard deviation sigma=339 was determined as for OQSO   ::
    :: by simulation.  (Sigma for OPSO, 290, is the true value (to   ::
    :: three places), not determined by simulation.                  ::
    :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
 OPSO test for generator block0.rng
  Output: No. missing words (mw), equiv normal variate (z), p-value (p)
                                                           mw     z     p
    OPSO for block0.rng      using bits 23 to 32        142244  1.154  .8758
    OPSO for block0.rng      using bits 22 to 31        142035   .433  .6676
    OPSO for block0.rng      using bits 21 to 30        141494 -1.432  .0760
    OPSO for block0.rng      using bits 20 to 29        141726  -.632  .2636
    OPSO for block0.rng      using bits 19 to 28        141593 -1.091  .1377
    OPSO for block0.rng      using bits 18 to 27        142055   .502  .6923
    OPSO for block0.rng      using bits 17 to 26        141315 -2.049  .0202
    OPSO for block0.rng      using bits 16 to 25        141424 -1.674  .0471
    OPSO for block0.rng      using bits 15 to 24        141745  -.567  .2855
    OPSO for block0.rng      using bits 14 to 23        141832  -.267  .3949
    OPSO for block0.rng      using bits 13 to 22        141643  -.918  .1792
    OPSO for block0.rng      using bits 12 to 21        141954   .154  .5612
    OPSO for block0.rng      using bits 11 to 20        141985   .261  .6029
    OPSO for block0.rng      using bits 10 to 19        141876  -.115  .4543
    OPSO for block0.rng      using bits  9 to 18        142271  1.247  .8938
    OPSO for block0.rng      using bits  8 to 17        141999   .309  .6214
    OPSO for block0.rng      using bits  7 to 16        141869  -.139  .4447
    OPSO for block0.rng      using bits  6 to 15        141857  -.180  .4284
    OPSO for block0.rng      using bits  5 to 14        141856  -.184  .4270
    OPSO for block0.rng      using bits  4 to 13        142131   .764  .7777
    OPSO for block0.rng      using bits  3 to 12        142168   .892  .8138
    OPSO for block0.rng      using bits  2 to 11        142080   .589  .7219
    OPSO for block0.rng      using bits  1 to 10        141612 -1.025  .1526
 OQSO test for generator block0.rng
  Output: No. missing words (mw), equiv normal variate (z), p-value (p)
                                                           mw     z     p
    OQSO for block0.rng      using bits 28 to 32        141672  -.805  .2106
    OQSO for block0.rng      using bits 27 to 31        141611 -1.011  .1559
    OQSO for block0.rng      using bits 26 to 30        142281  1.260  .8961
    OQSO for block0.rng      using bits 25 to 29        141774  -.459  .3232
    OQSO for block0.rng      using bits 24 to 28        142034   .423  .6637
    OQSO for block0.rng      using bits 23 to 27        142468  1.894  .9709
    OQSO for block0.rng      using bits 22 to 26        142100   .646  .7410
    OQSO for block0.rng      using bits 21 to 25        141932   .077  .5306
    OQSO for block0.rng      using bits 20 to 24        142181   .921  .8215
    OQSO for block0.rng      using bits 19 to 23        141312 -2.025  .0214
    OQSO for block0.rng      using bits 18 to 22        141852  -.194  .4230
    OQSO for block0.rng      using bits 17 to 21        142487  1.958  .9749
    OQSO for block0.rng      using bits 16 to 20        142074   .558  .7116
    OQSO for block0.rng      using bits 15 to 19        142282  1.263  .8968
    OQSO for block0.rng      using bits 14 to 18        142175   .901  .8161
```

```
   OQSO for block0.rng      using bits 13 to 17       141888  -.072  .4712
   OQSO for block0.rng      using bits 12 to 16       141469 -1.493  .0678
   OQSO for block0.rng      using bits 11 to 15       142293  1.301  .9033
   OQSO for block0.rng      using bits 10 to 14       142003   .318  .6246
   OQSO for block0.rng      using bits  9 to 13       141766  -.486  .3135
   OQSO for block0.rng      using bits  8 to 12       141867  -.143  .4430
   OQSO for block0.rng      using bits  7 to 11       142185   .934  .8250
   OQSO for block0.rng      using bits  6 to 10       142017   .365  .6424
   OQSO for block0.rng      using bits  5 to  9       142440  1.799  .9640
   OQSO for block0.rng      using bits  4 to  8       141971   .209  .5828
   OQSO for block0.rng      using bits  3 to  7       141716  -.655  .2561
   OQSO for block0.rng      using bits  2 to  6       141133 -2.632  .0042
   OQSO for block0.rng      using bits  1 to  5       142095   .629  .7355
  DNA test for generator block0.rng
  Output: No. missing words (mw), equiv normal variate (z), p-value (p)
                                                         mw     z     p
    DNA for block0.rng      using bits 31 to 32       142113   .601  .7260
    DNA for block0.rng      using bits 30 to 31       142436  1.554  .9399
    DNA for block0.rng      using bits 29 to 30       142500  1.742  .9593
    DNA for block0.rng      using bits 28 to 29       141793  -.343  .3657
    DNA for block0.rng      using bits 27 to 28       142059   .442  .6706
    DNA for block0.rng      using bits 26 to 27       141946   .108  .5431
    DNA for block0.rng      using bits 25 to 26       142110   .592  .7231
    DNA for block0.rng      using bits 24 to 25       142596  2.026  .9786
    DNA for block0.rng      using bits 23 to 24       142252  1.011  .8440
    DNA for block0.rng      using bits 22 to 23       142398  1.442  .9253
    DNA for block0.rng      using bits 21 to 22       141887  -.066  .4737
    DNA for block0.rng      using bits 20 to 21       141697  -.626  .2655
    DNA for block0.rng      using bits 19 to 20       141989   .235  .5929
    DNA for block0.rng      using bits 18 to 19       141340 -1.679  .0465
    DNA for block0.rng      using bits 17 to 18       142239   .972  .8346
    DNA for block0.rng      using bits 16 to 17       141424 -1.432  .0761
    DNA for block0.rng      using bits 15 to 16       142643  2.164  .9848
    DNA for block0.rng      using bits 14 to 15       141879  -.089  .4644
    DNA for block0.rng      using bits 13 to 14       142159   .736  .7693
    DNA for block0.rng      using bits 12 to 13       141770  -.411  .3405
    DNA for block0.rng      using bits 11 to 12       141704  -.606  .2724
    DNA for block0.rng      using bits 10 to 11       141889  -.060  .4761
    DNA for block0.rng      using bits  9 to 10       142157   .731  .7675
    DNA for block0.rng      using bits  8 to  9       141965   .164  .5652
    DNA for block0.rng      using bits  7 to  8       141840  -.205  .4190
    DNA for block0.rng      using bits  6 to  7       142055   .430  .6663
    DNA for block0.rng      using bits  5 to  6       142161   .742  .7711
    DNA for block0.rng      using bits  4 to  5       142031   .359  .6402
    DNA for block0.rng      using bits  3 to  4       142048   .409  .6588
    DNA for block0.rng      using bits  2 to  3       141338 -1.685  .0460
    DNA for block0.rng      using bits  1 to  2       141970   .179  .5710

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

    ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
    ::     This is the COUNT-THE-1's TEST on a stream of bytes.      ::
    :: Consider the file under test as a stream of bytes (four per   ::
    :: 32 bit integer).  Each byte can contain from 0 to 8 1's,      ::
    :: with probabilities 1,8,28,56,70,56,28,8,1 over 256.  Now let  ::
    :: the stream of bytes provide a string of overlapping  5-letter ::
    :: words, each "letter" taking values A,B,C,D,E. The letters are ::
```

```
    :: determined by the number of 1's in a byte::  0,1,or 2 yield A,::
    :: 3 yields B, 4 yields C, 5 yields D and 6,7 or 8 yield E. Thus ::
    :: we have a monkey at a typewriter hitting five keys with vari- ::
    :: ous probabilities (37,56,70,56,37 over 256).  There are 5^5   ::
    :: possible 5-letter words, and from a string of 256,000 (over-  ::
    :: lapping) 5-letter words, counts are made on the frequencies   ::
    :: for each word.   The quadratic form in the weak inverse of    ::
    :: the covariance matrix of the cell counts provides a chisquare ::
    :: test::  Q5-Q4, the difference of the naive Pearson sums of    ::
    :: (OBS-EXP)^2/EXP on counts for 5- and 4-letter cell counts.    ::
    ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
  Test results for block0.rng
Chi-square with 5^5-5^4=2500 d.of f. for sample size:2560000
                              chisquare  equiv normal  p-value
 Results fo COUNT-THE-1's in successive bytes:
byte stream for block0.rng       2446.69      -.754      .225440
byte stream for block0.rng       2534.80       .492      .688677

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

    ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
    ::     This is the COUNT-THE-1's TEST for specific bytes.        ::
    :: Consider the file under test as a stream of 32-bit integers.  ::
    :: From each integer, a specific byte is chosen , say the left-  ::
    :: most::  bits 1 to 8. Each byte can contain from 0 to 8 1's,   ::
    :: with probabilitie 1,8,28,56,70,56,28,8,1 over 256.  Now let   ::
    :: the specified bytes from successive integers provide a string ::
    :: of (overlapping) 5-letter words, each "letter" taking values  ::
    :: A,B,C,D,E. The letters are determined  by the number of 1's,  ::
    :: in that byte::  0,1,or 2 ---> A, 3 ---> B, 4 ---> C, 5 ---> D,::
    :: and  6,7 or 8 ---> E.  Thus we have a monkey at a typewriter  ::
    :: hitting five keys with with various probabilities::  37,56,70,::
    :: 56,37 over 256. There are 5^5 possible 5-letter words, and    ::
    :: from a string of 256,000 (overlapping) 5-letter words, counts ::
    :: are made on the frequencies for each word. The quadratic form ::
    :: in the weak inverse of the covariance matrix of the cell      ::
    :: counts provides a chisquare test::  Q5-Q4, the difference of  ::
    :: the naive Pearson  sums of (OBS-EXP)^2/EXP on counts for 5-   ::
    :: and 4-letter cell counts.                                     ::
    ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
Chi-square with 5^5-5^4=2500 d.of f. for sample size: 256000
                       chisquare  equiv normal  p value
 Results for COUNT-THE-1's in specified bytes:
          bits  1 to  8  2517.72       .251      .598921
          bits  2 to  9  2482.71      -.245      .403390
          bits  3 to 10  2576.00      1.075      .858760
          bits  4 to 11  2488.60      -.161      .435936
          bits  5 to 12  2513.93       .197      .578093
          bits  6 to 13  2552.13       .737      .769492
          bits  7 to 14  2562.49       .884      .811579
          bits  8 to 15  2508.38       .119      .547195
          bits  9 to 16  2481.69      -.259      .397849
          bits 10 to 17  2417.92     -1.161      .122852
          bits 11 to 18  2551.41       .727      .766406
          bits 12 to 19  2470.37      -.419      .337613
          bits 13 to 20  2525.04       .354      .638356
          bits 14 to 21  2490.29      -.137      .445373
```

```
        bits 15 to 22  2564.41       .911      .818810
        bits 16 to 23  2557.52       .813      .792012
        bits 17 to 24  2518.44       .261      .602889
        bits 18 to 25  2373.56     -1.788      .036877
        bits 19 to 26  2483.70      -.230      .408872
        bits 20 to 27  2532.05       .453      .674809
        bits 21 to 28  2480.42      -.277      .390920
        bits 22 to 29  2496.15      -.055      .478261
        bits 23 to 30  2489.79      -.144      .442617
        bits 24 to 31  2460.15      -.564      .286518
        bits 25 to 32  2599.73      1.410      .920780

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
     ::              THIS IS A PARKING LOT TEST                      ::
     :: In a square of side 100, randomly "park" a car---a circle of  ::
     :: radius 1.   Then try to park a 2nd, a 3rd, and so on, each    ::
     :: time parking "by ear".  That is, if an attempt to park a car  ::
     :: causes a crash with one already parked, try again at a new    ::
     :: random location. (To avoid path problems, consider parking    ::
     :: helicopters rather than cars.)   Each attempt leads to either ::
     :: a crash or a success, the latter followed by an increment to  ::
     :: the list of cars already parked. If we plot n:  the number of ::
     :: attempts, versus k::  the number successfully parked, we get a::
     :: curve that should be similar to those provided by a perfect   ::
     :: random number generator.  Theory for the behavior of such a   ::
     :: random curve seems beyond reach, and as graphics displays are ::
     :: not available for this battery of tests, a simple characteriz ::
     :: ation of the random experiment is used: k, the number of cars ::
     :: successfully parked after n=12,000 attempts. Simulation shows ::
     :: that k should average 3523 with sigma 21.9 and is very close  ::
     :: to normally distributed.  Thus (k-3523)/21.9 should be a st-  ::
     :: andard normal variable, which, converted to a uniform varia-  ::
     :: ble, provides input to a KSTEST based on a sample of 10.      ::
     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
           CDPARK: result of ten tests on file block0.rng
            Of 12,000 tries, the average no. of successes
                 should be 3523 with sigma=21.9
            Successes: 3532    z-score:   .411 p-value: .659449
            Successes: 3489    z-score: -1.553 p-value: .060270
            Successes: 3504    z-score:  -.868 p-value: .192812
            Successes: 3523    z-score:   .000 p-value: .500000
            Successes: 3517    z-score:  -.274 p-value: .392053
            Successes: 3528    z-score:   .228 p-value: .590298
            Successes: 3521    z-score:  -.091 p-value: .463618
            Successes: 3508    z-score:  -.685 p-value: .246694
            Successes: 3517    z-score:  -.274 p-value: .392053
            Successes: 3565    z-score:  1.918 p-value: .972432

           square size   avg. no.  parked   sample sigma
             100.            3520.400       19.012
            KSTEST for the above 10: p=  .300390

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
```

```
::               THE MINIMUM DISTANCE TEST                          ::
:: It does this 100 times::   choose n=8000 random points in a   ::
:: square of side 10000.  Find d, the minimum distance between   ::
:: the (n^2-n)/2 pairs of points.  If the points are truly inde- ::
:: pendent uniform, then d^2, the square of the minimum distance ::
:: should be (very close to) exponentially distributed with mean ::
:: .995 .  Thus 1-exp(-d^2/.995) should be uniform on [0,1) and  ::
:: a KSTEST on the resulting 100 values serves as a test of uni- ::
:: formity for random points in the square. Test numbers=0 mod 5 ::
:: are printed but the KSTEST is based on the full set of 100    ::
:: random choices of 8000 points in the 10000x10000 square.      ::
::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
         This is the MINIMUM DISTANCE test
        for random integers in the file block0.rng
Sample no.    d^2     avg     equiv uni
      5     .1127    .6202     .107114
     10     .3265    .5722     .279752
     15    1.5969    .8933     .799101
     20     .4793    .8284     .382297
     25     .0679    .7958     .065992
     30    3.1047    .9657     .955856
     35    1.0304   1.0735     .644983
     40     .3540   1.0087     .299374
     45    1.3808    .9832     .750369
     50     .9728    .9870     .623819
     55    4.6340   1.0585     .990508
     60     .8098   1.0668     .556858
     65    1.0949   1.0389     .667256
     70     .3421    .9843     .290915
     75     .0461    .9883     .045309
     80    1.3411    .9657     .740190
     85     .3548    .9493     .299955
     90     .5121   1.0054     .402281
     95     .5599    .9958     .430317
    100     .7538    .9921     .531210
MINIMUM DISTANCE TEST for block0.rng
     Result of KS test on 20 transformed mindist^2's:
                            p-value= .223584

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
::              THE 3DSPHERES TEST                                ::
:: Choose  4000 random points in a cube of edge 1000.  At each   ::
:: point, center a sphere large enough to reach the next closest ::
:: point. Then the volume of the smallest such sphere is (very   ::
:: close to) exponentially distributed with mean 120pi/3.  Thus  ::
:: the radius cubed is exponential with mean 30. (The mean is    ::
:: obtained by extensive simulation).  The 3DSPHERES test gener- ::
:: ates 4000 such spheres 20 times.  Each min radius cubed leads ::
:: to a uniform variable by means of 1-exp(-r^3/30.), then a     ::
::  KSTEST is done on the 20 p-values.                           ::
::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
         The 3DSPHERES test for file block0.rng
 sample no:  1     r^3=  23.673     p-value= .54575
 sample no:  2     r^3=   5.165     p-value= .15816
 sample no:  3     r^3=  38.008     p-value= .71830
```

```
 sample no:  4     r^3=  29.919     p-value= .63112
 sample no:  5     r^3=  19.700     p-value= .48142
 sample no:  6     r^3=  57.646     p-value= .85362
 sample no:  7     r^3=  14.649     p-value= .38632
 sample no:  8     r^3=  84.430     p-value= .94006
 sample no:  9     r^3=  10.395     p-value= .29283
 sample no: 10     r^3=  15.515     p-value= .40378
 sample no: 11     r^3=  20.948     p-value= .50255
 sample no: 12     r^3=  81.798     p-value= .93456
 sample no: 13     r^3=  16.334     p-value= .41984
 sample no: 14     r^3=  31.497     p-value= .65003
 sample no: 15     r^3=   3.279     p-value= .10354
 sample no: 16     r^3=   7.895     p-value= .23140
 sample no: 17     r^3=  37.448     p-value= .71299
 sample no: 18     r^3=  61.286     p-value= .87034
 sample no: 19     r^3=   5.064     p-value= .15533
 sample no: 20     r^3=  14.611     p-value= .38555
  A KS test is applied to those 20 p-values.
---------------------------------------------------------
       3DSPHERES test for file block0.rng           p-value= .133599
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

     :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
     ::      This is the SQEEZE test                                  ::
     ::  Random integers are floated to get uniforms on [0,1). Start- ::
     ::  ing with k=2^31=2147483647, the test finds j, the number of  ::
     ::  iterations necessary to reduce k to 1, using the reduction   ::
     ::  k=ceiling(k*U), with U provided by floating integers from    ::
     ::  the file being tested.  Such j's are found 100,000 times,    ::
     ::  then counts for the number of times j was <=6,7,...,47,>=48  ::
     ::  are used to provide a chi-square test for cell frequencies.  ::
     :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
           RESULTS OF SQUEEZE TEST FOR block0.rng
         Table of standardized frequency counts
     ( (obs-exp)/sqrt(exp) )^2
        for j taking values <=6,7,8,...,47,>=48:
     -.1     1.8     -.4    -1.1     1.1      .2
      .2      .8      .6     -.1     -.7     1.1
     1.1     1.0      .6      .0     -.9     -.8
     -.2     -.7      .0     -.9     -.4     -.6
     -.3     -.2      .2      .7     1.5      .6
     1.2     -.5      .0      .5      .1    -1.7
      .3      .8    -1.2    -1.8     -.6     1.0
     -.1
          Chi-square with 42 degrees of freedom: 29.384
             z-score= -1.377  p-value= .070700
______________________________________________________________

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

     :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
     ::             The  OVERLAPPING SUMS test                        ::
     :: Integers are floated to get a sequence U(1),U(2),... of uni-  ::
     :: form [0,1) variables.  Then overlapping sums,                 ::
     ::   S(1)=U(1)+...+U(100), S2=U(2)+...+U(101),... are formed.    ::
     :: The S's are virtually normal with a certain covariance mat-   ::
     :: rix.  A linear transformation of the S's converts them to a   ::
```

```
     :: sequence of independent standard normals, which are converted ::
     :: to uniform variables for a KSTEST. The  p-values from ten     ::
     :: KSTESTs are given still another KSTEST.                       ::
     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
                Test no.  1      p-value  .024581
                Test no.  2      p-value  .783819
                Test no.  3      p-value  .126377
                Test no.  4      p-value  .805804
                Test no.  5      p-value  .956854
                Test no.  6      p-value  .747899
                Test no.  7      p-value  .738569
                Test no.  8      p-value  .332539
                Test no.  9      p-value  .290589
                Test no. 10      p-value  .241737
   Results of the OSUM test for block0.rng
        KSTEST on the above 10 p-values:  .162389

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
     ::     This is the RUNS test.  It counts runs up, and runs down, ::
     :: in a sequence of uniform [0,1) variables, obtained by float-  ::
     :: ing the 32-bit integers in the specified file. This example   ::
     :: shows how runs are counted:  .123,.357,.789,.425,.224,.416,.95::
     :: contains an up-run of length 3, a down-run of length 2 and an ::
     :: up-run of (at least) 2, depending on the next values.  The    ::
     :: covariance matrices for the runs-up and runs-down are well    ::
     :: known, leading to chisquare tests for quadratic forms in the  ::
     :: weak inverses of the covariance matrices.  Runs are counted   ::
     :: for sequences of length 10,000.  This is done ten times. Then ::
     :: repeated.                                                     ::
     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
           The RUNS test for file block0.rng
     Up and down runs in a sample of 10000
_________________________________________________
                 Run test for block0.rng     :
       runs up; ks test for 10 p's: .895588
     runs down; ks test for 10 p's: .103022
                 Run test for block0.rng     :
       runs up; ks test for 10 p's: .689373
     runs down; ks test for 10 p's: .526976

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
     :: This is the CRAPS TEST. It plays 200,000 games of craps, finds::
     :: the number of wins and the number of throws necessary to end  ::
     :: each game.  The number of wins should be (very close to) a    ::
     :: normal with mean 200000p and variance 200000p(1-p), with      ::
     :: p=244/495.  Throws necessary to complete the game can vary    ::
     :: from 1 to infinity, but counts for all>21 are lumped with 21. ::
     :: A chi-square test is made on the no.-of-throws cell counts.   ::
     :: Each 32-bit integer from the test file provides the value for ::
     :: the throw of a die, by floating to [0,1), multiplying by 6    ::
     :: and taking 1 plus the integer part of the result.             ::
     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
                Results of craps test for block0.rng
```

```
 No. of wins:  Observed Expected
                            98204    98585.86
                98204= No. of wins, z-score=-1.708 pvalue= .04383
  Analysis of Throws-per-Game:
Chisq=  23.90 for 20 degrees of freedom, p=  .75316
             Throws Observed Expected  Chisq     Sum
                1    66462    66666.7    .628     .628
                2    37635    37654.3    .010     .638
                3    26885    26954.7    .180     .819
                4    19335    19313.5    .024     .843
                5    13990    13851.4   1.386    2.229
                6    10053     9943.5   1.205    3.434
                7     7095     7145.0    .350    3.784
                8     5140     5139.1    .000    3.784
                9     3733     3699.9    .297    4.081
               10     2769     2666.3   3.956    8.037
               11     1904     1923.3    .194    8.232
               12     1364     1388.7    .441    8.672
               13     1038     1003.7   1.171    9.843
               14      715      726.1    .171   10.014
               15      508      525.8    .605   10.619
               16      404      381.2   1.370   11.989
               17      258      276.5   1.243   13.232
               18      163      200.8   7.126   20.358
               19      156      146.0    .687   21.045
               20      121      106.2   2.058   23.103
               21      272      287.1    .796   23.899
          SUMMARY  FOR block0.rng
              p-value for no. of wins: .043828
              p-value for throws/game: .753159

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

Results of DIEHARD battery of tests sent to file report0.txt



      NOTE: Most of the tests in DIEHARD return a p-value, which
      should be uniform on [0,1) if the input file contains truly
      independent random bits.   Those p-values are obtained by
      p=F(X), where F is the assumed distribution of the sample
      random variable X---often normal. But that assumed F is just
      an asymptotic approximation, for which the fit will be worst
      in the tails. Thus you should not be surprised with
      occasional p-values near 0 or 1, such as .0012 or .9983.
      When a bit stream really FAILS BIG, you will get p's of 0 or
      1 to six or more places.  By all means, do not, as a
      Statistician might, think that a p < .025 or p> .975 means
      that the RNG has "failed the test at the .05 level".  Such
      p's happen among the hundreds that DIEHARD produces, even
      with good RNG's.  So keep in mind that " p happens".
    ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
    ::            This is the BIRTHDAY SPACINGS TEST                ::
    :: Choose m birthdays in a year of n days.  List the spacings    ::
    :: between the birthdays.  If j is the number of values that     ::
    :: occur more than once in that list, then j is asymptotically   ::
```

```
    :: Poisson distributed with mean m^3/(4n).  Experience shows n   ::
    :: must be quite large, say n>=2^18, for comparing the results   ::
    :: to the Poisson distribution with that mean.  This test uses   ::
    :: n=2^24 and m=2^9,  so that the underlying distribution for j  ::
    :: is taken to be Poisson with lambda=2^27/(2^26)=2.  A sample   ::
    :: of 500 j's is taken, and a chi-square goodness of fit test    ::
    :: provides a p value.  The first test uses bits 1-24 (counting  ::
    :: from the left) from integers in the specified file.           ::
    ::   Then the file is closed and reopened. Next, bits 2-25 are   ::
    :: used to provide birthdays, then 3-26 and so on to bits 9-32.  ::
    :: Each set of bits provides a p-value, and the nine p-values    ::
    :: provide a sample for a KSTEST.                                ::
    ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
 BIRTHDAY SPACINGS TEST, M= 512 N=2**24 LAMBDA=  2.0000
           Results for block1.rng
                   For a sample of size 500:     mean
           block1.rng      using bits  1 to 24   2.016
  duplicate       number       number
  spacings       observed     expected
        0          56.       67.668
        1         145.      135.335
        2         141.      135.335
        3          93.       90.224
        4          40.       45.112
        5          12.       18.045
  6 to INF         13.        8.282
 Chisquare with  6 d.o.f. =     8.32 p-value=  .784188
  :::::::::::::::::::::::::::::::::::::::::
                   For a sample of size 500:     mean
           block1.rng      using bits  2 to 25   2.038
  duplicate       number       number
  spacings       observed     expected
        0          66.       67.668
        1         132.      135.335
        2         140.      135.335
        3          83.       90.224
        4          51.       45.112
        5          18.       18.045
  6 to INF         10.        8.282
 Chisquare with  6 d.o.f. =     1.99 p-value=  .079160
  :::::::::::::::::::::::::::::::::::::::::
                   For a sample of size 500:     mean
           block1.rng      using bits  3 to 26   1.930
  duplicate       number       number
  spacings       observed     expected
        0          73.       67.668
        1         148.      135.335
        2         145.      135.335
        3          61.       90.224
        4          42.       45.112
        5          18.       18.045
  6 to INF         13.        8.282
 Chisquare with  6 d.o.f. =    14.66 p-value=  .976961
  :::::::::::::::::::::::::::::::::::::::::
                   For a sample of size 500:     mean
           block1.rng      using bits  4 to 27   1.912
  duplicate       number       number
```

```
 spacings       observed     expected
      0           81.       67.668
      1          132.      135.335
      2          141.      135.335
      3           76.       90.224
      4           48.       45.112
      5           14.       18.045
 6 to INF          8.        8.282
Chisquare with  6 d.o.f. =     6.29 p-value=  .608453
 :::::::::::::::::::::::::::::::::::::::::
                  For a sample of size 500:     mean
          block1.rng      using bits  5 to 28   1.996
 duplicate       number       number
 spacings       observed     expected
      0           63.       67.668
      1          136.      135.335
      2          141.      135.335
      3           88.       90.224
      4           52.       45.112
      5           13.       18.045
 6 to INF          7.        8.282
Chisquare with  6 d.o.f. =     3.28 p-value=  .226722
 :::::::::::::::::::::::::::::::::::::::::
                  For a sample of size 500:     mean
          block1.rng      using bits  6 to 29   2.016
 duplicate       number       number
 spacings       observed     expected
      0           60.       67.668
      1          133.      135.335
      2          148.      135.335
      3           89.       90.224
      4           45.       45.112
      5           19.       18.045
 6 to INF          6.        8.282
Chisquare with  6 d.o.f. =     2.79 p-value=  .165348
 :::::::::::::::::::::::::::::::::::::::::
                  For a sample of size 500:     mean
          block1.rng      using bits  7 to 30   1.898
 duplicate       number       number
 spacings       observed     expected
      0           88.       67.668
      1          127.      135.335
      2          122.      135.335
      3           92.       90.224
      4           56.       45.112
      5           13.       18.045
 6 to INF          2.        8.282
Chisquare with  6 d.o.f. =    16.77 p-value=  .989853
 :::::::::::::::::::::::::::::::::::::::::
                  For a sample of size 500:     mean
          block1.rng      using bits  8 to 31   2.052
 duplicate       number       number
 spacings       observed     expected
      0           66.       67.668
      1          126.      135.335
      2          145.      135.335
      3           91.       90.224
```

```
       4          37.        45.112
       5          23.        18.045
 6 to INF         12.         8.282
Chisquare with  6 d.o.f. =     5.87 p-value=  .562162
 :::::::::::::::::::::::::::::::::::::::::
                 For a sample of size 500:     mean
         block1.rng      using bits  9 to 32   1.940
 duplicate       number       number
 spacings       observed     expected
       0          73.        67.668
       1         141.       135.335
       2         130.       135.335
       3          96.        90.224
       4          34.        45.112
       5          14.        18.045
 6 to INF         12.         8.282
Chisquare with  6 d.o.f. =     6.55 p-value=  .635570
 :::::::::::::::::::::::::::::::::::::::::
  The 9 p-values were
       .784188   .079160   .976961   .608453   .226722
       .165348   .989853   .562162   .635570
 A KSTEST for the 9 p-values yields  .514385

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

    ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
    ::            THE OVERLAPPING 5-PERMUTATION TEST                  ::
    :: This is the OPERM5 test.  It looks at a sequence of one mill- ::
    :: ion 32-bit random integers.  Each set of five consecutive     ::
    :: integers can be in one of 120 states, for the 5! possible or- ::
    :: derings of five numbers.  Thus the 5th, 6th, 7th,...numbers   ::
    :: each provide a state. As many thousands of state transitions  ::
    :: are observed,  cumulative counts are made of the number of    ::
    :: occurences of each state.  Then the quadratic form in the     ::
    :: weak inverse of the 120x120 covariance matrix yields a test   ::
    :: equivalent to the likelihood ratio test that the 120 cell     ::
    :: counts came from the specified (asymptotically) normal dis-   ::
    :: tribution with the specified 120x120 covariance matrix (with  ::
    :: rank 99).  This version uses 1,000,000 integers, twice.       ::
    ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
          OPERM5 test for file block1.rng
    For a sample of 1,000,000 consecutive 5-tuples,
chisquare for 99 degrees of freedom=111.433; p-value= .814967
          OPERM5 test for file block1.rng
    For a sample of 1,000,000 consecutive 5-tuples,
chisquare for 99 degrees of freedom=102.642; p-value= .619097
    ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
    :: This is the BINARY RANK TEST for 31x31 matrices. The leftmost ::
    :: 31 bits of 31 random integers from the test sequence are used ::
    :: to form a 31x31 binary matrix over the field {0,1}. The rank  ::
    :: is determined. That rank can be from 0 to 31, but ranks< 28   ::
    :: are rare, and their counts are pooled with those for rank 28. ::
    :: Ranks are found for 40,000 such random matrices and a chisqua-::
    :: re test is performed on counts for ranks 31,30,29 and <=28.   ::
    ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
   Binary rank test for block1.rng
        Rank test for 31x31 binary matrices:
```

```
        rows from leftmost 31 bits of each 32-bit integer
      rank   observed  expected (o-e)^2/e  sum
        28       208     211.4   .055259     .055
        29      5183    5134.0   .467470     .523
        30     23037   23103.0   .188814     .712
        31     11572   11551.5   .036294     .748
  chisquare=  .748 for 3 d. of f.; p-value= .331683
--------------------------------------------------------------
     :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
     :: This is the BINARY RANK TEST for 32x32 matrices. A random 32x ::
     :: 32 binary matrix is formed, each row a 32-bit random integer. ::
     :: The rank is determined. That rank can be from 0 to 32, ranks  ::
     :: less than 29 are rare, and their counts are pooled with those ::
     :: for rank 29.  Ranks are found for 40,000 such random matrices ::
     :: and a chisquare test is performed on counts for ranks  32,31, ::
     :: 30 and <=29.                                                  ::
     :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
    Binary rank test for block1.rng
         Rank test for 32x32 binary matrices:
        rows from leftmost 32 bits of each 32-bit integer
      rank   observed  expected (o-e)^2/e  sum
        29       216     211.4   .099304     .099
        30      5040    5134.0  1.721447    1.821
        31     23289   23103.0  1.496710    3.317
        32     11455   11551.5   .806557    4.124
  chisquare= 4.124 for 3 d. of f.; p-value= .772759
--------------------------------------------------------------

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

     :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
     :: This is the BINARY RANK TEST for 6x8 matrices.  From each of  ::
     :: six random 32-bit integers from the generator under test, a   ::
     :: specified byte is chosen, and the resulting six bytes form a  ::
     :: 6x8 binary matrix whose rank is determined.  That rank can be ::
     :: from 0 to 6, but ranks 0,1,2,3 are rare; their counts are     ::
     :: pooled with those for rank 4. Ranks are found for 100,000     ::
     :: random matrices, and a chi-square test is performed on       ::
     :: counts for ranks 6,5 and <=4.                                 ::
     :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
         Binary Rank Test for block1.rng
        Rank of a 6x8 binary matrix,
     rows formed from eight bits of the RNG block1.rng
     b-rank test for bits  1 to  8
                     OBSERVED   EXPECTED     (O-E)^2/E      SUM
          r<=4          949       944.3        .023        .023
          r =5        21640     21743.9        .496        .520
          r =6        77411     77311.8        .127        .647
                        p=1-exp(-SUM/2)= .27643
        Rank of a 6x8 binary matrix,
     rows formed from eight bits of the RNG block1.rng
     b-rank test for bits  2 to  9
                     OBSERVED   EXPECTED     (O-E)^2/E      SUM
          r<=4          961       944.3        .295        .295
          r =5        21546     21743.9       1.801       2.096
          r =6        77493     77311.8        .425       2.521
                        p=1-exp(-SUM/2)= .71651
```

```
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block1.rng
b-rank test for bits  3 to 10
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          953       944.3         .080         .080
     r =5        21567     21743.9        1.439        1.519
     r =6        77480     77311.8         .366        1.885
                   p=1-exp(-SUM/2)= .61040
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block1.rng
b-rank test for bits  4 to 11
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          962       944.3         .332         .332
     r =5        21532     21743.9        2.065        2.397
     r =6        77506     77311.8         .488        2.885
                   p=1-exp(-SUM/2)= .76361
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block1.rng
b-rank test for bits  5 to 12
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          950       944.3         .034         .034
     r =5        21664     21743.9         .294         .328
     r =6        77386     77311.8         .071         .399
                   p=1-exp(-SUM/2)= .18094
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block1.rng
b-rank test for bits  6 to 13
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          937       944.3         .056         .056
     r =5        21659     21743.9         .331         .388
     r =6        77404     77311.8         .110         .498
                   p=1-exp(-SUM/2)= .22038
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block1.rng
b-rank test for bits  7 to 14
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          901       944.3        1.986        1.986
     r =5        21810     21743.9         .201        2.187
     r =6        77289     77311.8         .007        2.193
                   p=1-exp(-SUM/2)= .66601
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block1.rng
b-rank test for bits  8 to 15
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          932       944.3         .160         .160
     r =5        21701     21743.9         .085         .245
     r =6        77367     77311.8         .039         .284
                   p=1-exp(-SUM/2)= .13251
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block1.rng
b-rank test for bits  9 to 16
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          978       944.3        1.203        1.203
     r =5        21771     21743.9         .034        1.236
     r =6        77251     77311.8         .048        1.284
                   p=1-exp(-SUM/2)= .47381
   Rank of a 6x8 binary matrix,
```

```
rows formed from eight bits of the RNG block1.rng
b-rank test for bits 10 to 17
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4         1026       944.3       7.068        7.068
     r =5        21771     21743.9        .034        7.102
     r =6        77203     77311.8        .153        7.255
                   p=1-exp(-SUM/2)= .97342
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block1.rng
b-rank test for bits 11 to 18
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          959       944.3        .229         .229
     r =5        21803     21743.9        .161         .389
     r =6        77238     77311.8        .070         .460
                   p=1-exp(-SUM/2)= .20542
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block1.rng
b-rank test for bits 12 to 19
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          986       944.3       1.841        1.841
     r =5        21643     21743.9        .468        2.310
     r =6        77371     77311.8        .045        2.355
                   p=1-exp(-SUM/2)= .69193
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block1.rng
b-rank test for bits 13 to 20
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          991       944.3       2.309        2.309
     r =5        21666     21743.9        .279        2.588
     r =6        77343     77311.8        .013        2.601
                   p=1-exp(-SUM/2)= .72762
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block1.rng
b-rank test for bits 14 to 21
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          965       944.3        .454         .454
     r =5        21684     21743.9        .165         .619
     r =6        77351     77311.8        .020         .639
                   p=1-exp(-SUM/2)= .27334
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block1.rng
b-rank test for bits 15 to 22
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          919       944.3        .678         .678
     r =5        21708     21743.9        .059         .737
     r =6        77373     77311.8        .048         .786
                   p=1-exp(-SUM/2)= .32484
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block1.rng
b-rank test for bits 16 to 23
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          900       944.3       2.078        2.078
     r =5        21586     21743.9       1.147        3.225
     r =6        77514     77311.8        .529        3.754
                   p=1-exp(-SUM/2)= .84694
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block1.rng
```

```
b-rank test for bits 17 to 24
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          914       944.3         .972         .972
     r =5        21643     21743.9         .468        1.441
     r =6        77443     77311.8         .223        1.663
                   p=1-exp(-SUM/2)= .56464
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block1.rng
b-rank test for bits 18 to 25
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          989       944.3        2.116        2.116
     r =5        21855     21743.9         .568        2.683
     r =6        77156     77311.8         .314        2.997
                   p=1-exp(-SUM/2)= .77659
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block1.rng
b-rank test for bits 19 to 26
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4         1032       944.3        8.145        8.145
     r =5        21790     21743.9         .098        8.242
     r =6        77178     77311.8         .232        8.474
                   p=1-exp(-SUM/2)= .98555
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block1.rng
b-rank test for bits 20 to 27
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          939       944.3         .030         .030
     r =5        21558     21743.9        1.589        1.619
     r =6        77503     77311.8         .473        2.092
                   p=1-exp(-SUM/2)= .64865
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block1.rng
b-rank test for bits 21 to 28
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          964       944.3         .411         .411
     r =5        21617     21743.9         .741        1.152
     r =6        77419     77311.8         .149        1.300
                   p=1-exp(-SUM/2)= .47800
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block1.rng
b-rank test for bits 22 to 29
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          947       944.3         .008         .008
     r =5        21627     21743.9         .628         .636
     r =6        77426     77311.8         .169         .805
                   p=1-exp(-SUM/2)= .33131
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block1.rng
b-rank test for bits 23 to 30
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          939       944.3         .030         .030
     r =5        21854     21743.9         .557         .587
     r =6        77207     77311.8         .142         .729
                   p=1-exp(-SUM/2)= .30557
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block1.rng
b-rank test for bits 24 to 31
```

```
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4         1010       944.3       4.571        4.571
     r =5        21818     21743.9        .253        4.823
     r =6        77172     77311.8        .253        5.076
                   p=1-exp(-SUM/2)= .92099
   Rank of a 6x8 binary matrix,
  rows formed from eight bits of the RNG block1.rng
  b-rank test for bits 25 to 32
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          924       944.3        .436         .436
     r =5        21847     21743.9        .489         .925
     r =6        77229     77311.8        .089        1.014
                   p=1-exp(-SUM/2)= .39770
  TEST SUMMARY, 25 tests on 100,000 random 6x8 matrices
 These should be 25 uniform [0,1] random variables:
     .276434     .716507     .610395     .763609     .180941
     .220379     .666005     .132506     .473809     .973421
     .205421     .691935     .727615     .273340     .324845
     .846937     .564643     .776588     .985549     .648652
     .477998     .331311     .305568     .920986     .397697
   brank test summary for block1.rng
     The KS test for those 25 supposed UNI's yields
               KS p-value= .414012

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

     :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
     ::                  THE BITSTREAM TEST                           ::
     :: The file under test is viewed as a stream of bits. Call them  ::
     :: b1,b2,... .  Consider an alphabet with two "letters", 0 and 1 ::
     :: and think of the stream of bits as a succession of 20-letter  ::
     :: "words", overlapping.  Thus the first word is b1b2...b20, the ::
     :: second is b2b3...b21, and so on.  The bitstream test counts   ::
     :: the number of missing 20-letter (20-bit) words in a string of ::
     :: 2^21 overlapping 20-letter words.  There are 2^20 possible 20 ::
     :: letter words.  For a truly random string of 2^21+19 bits, the ::
     :: number of missing words j should be (very close to) normally  ::
     :: distributed with mean 141,909 and sigma 428.  Thus            ::
     ::  (j-141909)/428 should be a standard normal variate (z score) ::
     :: that leads to a uniform [0,1) p value.  The test is repeated  ::
     :: twenty times.                                                 ::
     :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
 THE OVERLAPPING 20-tuples BITSTREAM  TEST, 20 BITS PER WORD, N words
   This test uses N=2^21 and samples the bitstream 20 times.
  No. missing words should average  141909. with sigma=428.
----------------------------------------------------------
 tst no  1:  142400 missing words,    1.15 sigmas from mean, p-value= .87419
 tst no  2:  141534 missing words,    -.88 sigmas from mean, p-value= .19026
 tst no  3:  142040 missing words,     .31 sigmas from mean, p-value= .61993
 tst no  4:  141799 missing words,    -.26 sigmas from mean, p-value= .39829
 tst no  5:  142154 missing words,     .57 sigmas from mean, p-value= .71622
 tst no  6:  141972 missing words,     .15 sigmas from mean, p-value= .55821
 tst no  7:  141667 missing words,    -.57 sigmas from mean, p-value= .28563
 tst no  8:  141822 missing words,    -.20 sigmas from mean, p-value= .41916
 tst no  9:  142618 missing words,    1.66 sigmas from mean, p-value= .95112
 tst no 10:  142531 missing words,    1.45 sigmas from mean, p-value= .92682
 tst no 11:  142052 missing words,     .33 sigmas from mean, p-value= .63056
```

```
 tst no 12:  141780 missing words,    -.30 sigmas from mean, p-value= .38126
 tst no 13:  141425 missing words,   -1.13 sigmas from mean, p-value= .12890
 tst no 14:  141711 missing words,    -.46 sigmas from mean, p-value= .32154
 tst no 15:  141938 missing words,     .07 sigmas from mean, p-value= .52671
 tst no 16:  141950 missing words,     .10 sigmas from mean, p-value= .53785
 tst no 17:  141412 missing words,   -1.16 sigmas from mean, p-value= .12262
 tst no 18:  141698 missing words,    -.49 sigmas from mean, p-value= .31074
 tst no 19:  142031 missing words,     .28 sigmas from mean, p-value= .61190
 tst no 20:  141678 missing words,    -.54 sigmas from mean, p-value= .29443

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

     :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
     ::             The tests OPSO, OQSO and DNA                      ::
     ::         OPSO means Overlapping-Pairs-Sparse-Occupancy         ::
     :: The OPSO test considers 2-letter words from an alphabet of    ::
     :: 1024 letters.  Each letter is determined by a specified ten   ::
     :: bits from a 32-bit integer in the sequence to be tested. OPSO ::
     :: generates  2^21 (overlapping) 2-letter words  (from 2^21+1    ::
     :: "keystrokes")  and counts the number of missing words---that  ::
     :: is 2-letter words which do not appear in the entire sequence. ::
     :: That count should be very close to normally distributed with  ::
     :: mean 141,909, sigma 290. Thus (missingwrds-141909)/290 should ::
     :: be a standard normal variable. The OPSO test takes 32 bits at ::
     :: a time from the test file and uses a designated set of ten    ::
     :: consecutive bits. It then restarts the file for the next de-  ::
     :: signated 10 bits, and so on.                                  ::
     ::                                                               ::
     ::     OQSO means Overlapping-Quadruples-Sparse-Occupancy        ::
     ::   The test OQSO is similar, except that it considers 4-letter ::
     :: words from an alphabet of 32 letters, each letter determined  ::
     :: by a designated string of 5 consecutive bits from the test    ::
     :: file, elements of which are assumed 32-bit random integers.   ::
     :: The mean number of missing words in a sequence of 2^21 four-  ::
     :: letter words,  (2^21+3 "keystrokes"), is again 141909, with   ::
     :: sigma = 295.  The mean is based on theory; sigma comes from   ::
     :: extensive simulation.                                         ::
     ::                                                               ::
     ::    The DNA test considers an alphabet of 4 letters::  C,G,A,T,::
     :: determined by two designated bits in the sequence of random   ::
     :: integers being tested.  It considers 10-letter words, so that ::
     :: as in OPSO and OQSO, there are 2^20 possible words, and the   ::
     :: mean number of missing words from a string of 2^21  (over-    ::
     :: lapping)  10-letter  words (2^21+9 "keystrokes") is 141909.   ::
     :: The standard deviation sigma=339 was determined as for OQSO   ::
     :: by simulation.  (Sigma for OPSO, 290, is the true value (to   ::
     :: three places), not determined by simulation.                  ::
     :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
 OPSO test for generator block1.rng
  Output: No. missing words (mw), equiv normal variate (z), p-value (p)
                                                           mw     z     p
    OPSO for block1.rng      using bits 23 to 32        141950   .140  .5558
    OPSO for block1.rng      using bits 22 to 31        141581 -1.132  .1288
    OPSO for block1.rng      using bits 21 to 30        141494 -1.432  .0760
    OPSO for block1.rng      using bits 20 to 29        142281  1.282  .9000
    OPSO for block1.rng      using bits 19 to 28        141780  -.446  .3278
    OPSO for block1.rng      using bits 18 to 27        142163   .875  .8091
```

```
   OPSO for block1.rng      using bits 17 to 26        142118   .720  .7641
   OPSO for block1.rng      using bits 16 to 25        141461 -1.546  .0611
   OPSO for block1.rng      using bits 15 to 24        141481 -1.477  .0698
   OPSO for block1.rng      using bits 14 to 23        142264  1.223  .8893
   OPSO for block1.rng      using bits 13 to 22        141703  -.711  .2384
   OPSO for block1.rng      using bits 12 to 21        141614 -1.018  .1543
   OPSO for block1.rng      using bits 11 to 20        142151   .833  .7977
   OPSO for block1.rng      using bits 10 to 19        141895  -.049  .4803
   OPSO for block1.rng      using bits  9 to 18        141928   .064  .5257
   OPSO for block1.rng      using bits  8 to 17        142159   .861  .8054
   OPSO for block1.rng      using bits  7 to 16        142062   .526  .7007
   OPSO for block1.rng      using bits  6 to 15        141493 -1.436  .0756
   OPSO for block1.rng      using bits  5 to 14        141534 -1.294  .0978
   OPSO for block1.rng      using bits  4 to 13        142510  2.071  .9808
   OPSO for block1.rng      using bits  3 to 12        141904  -.018  .4927
   OPSO for block1.rng      using bits  2 to 11        141845  -.222  .4122
   OPSO for block1.rng      using bits  1 to 10        141676  -.805  .2105
OQSO test for generator block1.rng
 Output: No. missing words (mw), equiv normal variate (z), p-value (p)
                                                          mw     z     p
   OQSO for block1.rng      using bits 28 to 32        142029   .406  .6575
   OQSO for block1.rng      using bits 27 to 31        141617  -.991  .1609
   OQSO for block1.rng      using bits 26 to 30        142139   .779  .7819
   OQSO for block1.rng      using bits 25 to 29        141704  -.696  .2432
   OQSO for block1.rng      using bits 24 to 28        142564  2.219  .9868
   OQSO for block1.rng      using bits 23 to 27        142138   .775  .7809
   OQSO for block1.rng      using bits 22 to 26        142211  1.023  .8468
   OQSO for block1.rng      using bits 21 to 25        141872  -.127  .4497
   OQSO for block1.rng      using bits 20 to 24        142263  1.199  .8847
   OQSO for block1.rng      using bits 19 to 23        141500 -1.388  .0826
   OQSO for block1.rng      using bits 18 to 22        141806  -.350  .3631
   OQSO for block1.rng      using bits 17 to 21        141569 -1.154  .1243
   OQSO for block1.rng      using bits 16 to 20        141600 -1.049  .1472
   OQSO for block1.rng      using bits 15 to 19        141782  -.432  .3330
   OQSO for block1.rng      using bits 14 to 18        142098   .640  .7388
   OQSO for block1.rng      using bits 13 to 17        141757  -.516  .3028
   OQSO for block1.rng      using bits 12 to 16        142065   .528  .7011
   OQSO for block1.rng      using bits 11 to 15        141791  -.401  .3442
   OQSO for block1.rng      using bits 10 to 14        142234  1.101  .8645
   OQSO for block1.rng      using bits  9 to 13        141474 -1.476  .0700
   OQSO for block1.rng      using bits  8 to 12        142218  1.046  .8523
   OQSO for block1.rng      using bits  7 to 11        141723  -.632  .2638
   OQSO for block1.rng      using bits  6 to 10        141865  -.150  .4403
   OQSO for block1.rng      using bits  5 to  9        141451 -1.554  .0601
   OQSO for block1.rng      using bits  4 to  8        141919   .033  .5131
   OQSO for block1.rng      using bits  3 to  7        142509  2.033  .9790
   OQSO for block1.rng      using bits  2 to  6        141629  -.950  .1710
   OQSO for block1.rng      using bits  1 to  5        141846  -.215  .4150
 DNA test for generator block1.rng
 Output: No. missing words (mw), equiv normal variate (z), p-value (p)
                                                          mw     z     p
    DNA for block1.rng      using bits 31 to 32        141889  -.060  .4761
    DNA for block1.rng      using bits 30 to 31        141862  -.140  .4445
    DNA for block1.rng      using bits 29 to 30        141929   .058  .5231
    DNA for block1.rng      using bits 28 to 29        141466 -1.308  .0955
    DNA for block1.rng      using bits 27 to 28        141964   .161  .5641
    DNA for block1.rng      using bits 26 to 27        142123   .630  .7358
```

```
     DNA for block1.rng      using bits 25 to 26        141814  -.281  .3893
     DNA for block1.rng      using bits 24 to 25        141495 -1.222  .1108
     DNA for block1.rng      using bits 23 to 24        142149   .707  .7602
     DNA for block1.rng      using bits 22 to 23        141730  -.529  .2984
     DNA for block1.rng      using bits 21 to 22        142132   .657  .7444
     DNA for block1.rng      using bits 20 to 21        142298  1.147  .8742
     DNA for block1.rng      using bits 19 to 20        142103   .571  .7161
     DNA for block1.rng      using bits 18 to 19        142308  1.176  .8802
     DNA for block1.rng      using bits 17 to 18        141821  -.261  .3972
     DNA for block1.rng      using bits 16 to 17        142027   .347  .6357
     DNA for block1.rng      using bits 15 to 16        141820  -.264  .3961
     DNA for block1.rng      using bits 14 to 15        141978   .203  .5803
     DNA for block1.rng      using bits 13 to 14        142212   .893  .8140
     DNA for block1.rng      using bits 12 to 13        141838  -.210  .4167
     DNA for block1.rng      using bits 11 to 12        142587  1.999  .9772
     DNA for block1.rng      using bits 10 to 11        141981   .211  .5837
     DNA for block1.rng      using bits  9 to 10        141678  -.682  .2475
     DNA for block1.rng      using bits  8 to  9        141288 -1.833  .0334
     DNA for block1.rng      using bits  7 to  8        141794  -.340  .3669
     DNA for block1.rng      using bits  6 to  7        141596  -.924  .1777
     DNA for block1.rng      using bits  5 to  6        141716  -.570  .2842
     DNA for block1.rng      using bits  4 to  5        142093   .542  .7060
     DNA for block1.rng      using bits  3 to  4        142198   .852  .8028
     DNA for block1.rng      using bits  2 to  3        141994   .250  .5986
     DNA for block1.rng      using bits  1 to  2        141433 -1.405  .0800

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

     :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
     ::     This is the COUNT-THE-1's TEST on a stream of bytes.      ::
     :: Consider the file under test as a stream of bytes (four per   ::
     :: 32 bit integer).  Each byte can contain from 0 to 8 1's,      ::
     :: with probabilities 1,8,28,56,70,56,28,8,1 over 256.  Now let  ::
     :: the stream of bytes provide a string of overlapping  5-letter ::
     :: words, each "letter" taking values A,B,C,D,E. The letters are ::
     :: determined by the number of 1's in a byte::  0,1,or 2 yield A,::
     :: 3 yields B, 4 yields C, 5 yields D and 6,7 or 8 yield E. Thus ::
     :: we have a monkey at a typewriter hitting five keys with vari- ::
     :: ous probabilities (37,56,70,56,37 over 256).  There are 5^5   ::
     :: possible 5-letter words, and from a string of 256,000 (over-  ::
     :: lapping) 5-letter words, counts are made on the frequencies   ::
     :: for each word.   The quadratic form in the weak inverse of    ::
     :: the covariance matrix of the cell counts provides a chisquare ::
     :: test::  Q5-Q4, the difference of the naive Pearson sums of    ::
     :: (OBS-EXP)^2/EXP on counts for 5- and 4-letter cell counts.    ::
     :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
   Test results for block1.rng
 Chi-square with 5^5-5^4=2500 d.of f. for sample size:2560000
                              chisquare  equiv normal  p-value
  Results fo COUNT-THE-1's in successive bytes:
 byte stream for block1.rng       2564.22       .908      .818123
 byte stream for block1.rng       2455.82      -.625      .266035

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

     :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
     ::     This is the COUNT-THE-1's TEST for specific bytes.        ::
```

```
     :: Consider the file under test as a stream of 32-bit integers.  ::
     :: From each integer, a specific byte is chosen , say the left-  ::
     :: most::  bits 1 to 8. Each byte can contain from 0 to 8 1's,   ::
     :: with probabilitie 1,8,28,56,70,56,28,8,1 over 256.  Now let   ::
     :: the specified bytes from successive integers provide a string ::
     :: of (overlapping) 5-letter words, each "letter" taking values  ::
     :: A,B,C,D,E. The letters are determined  by the number of 1's,  ::
     :: in that byte::  0,1,or 2 ---> A, 3 ---> B, 4 ---> C, 5 ---> D,::
     :: and  6,7 or 8 ---> E.  Thus we have a monkey at a typewriter  ::
     :: hitting five keys with with various probabilities::  37,56,70,::
     :: 56,37 over 256. There are 5^5 possible 5-letter words, and    ::
     :: from a string of 256,000 (overlapping) 5-letter words, counts ::
     :: are made on the frequencies for each word. The quadratic form ::
     :: in the weak inverse of the covariance matrix of the cell      ::
     :: counts provides a chisquare test::  Q5-Q4, the difference of  ::
     :: the naive Pearson  sums of (OBS-EXP)^2/EXP on counts for 5-   ::
     :: and 4-letter cell counts.                                     ::
     :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
 Chi-square with 5^5-5^4=2500 d.of f. for sample size: 256000
                     chisquare  equiv normal  p value
  Results for COUNT-THE-1's in specified bytes:
           bits  1 to  8  2406.97     -1.316      .094158
           bits  2 to  9  2554.76       .774      .780651
           bits  3 to 10  2421.51     -1.110      .133495
           bits  4 to 11  2638.13      1.953      .974619
           bits  5 to 12  2575.88      1.073      .858396
           bits  6 to 13  2464.64      -.500      .308537
           bits  7 to 14  2351.08     -2.106      .017601
           bits  8 to 15  2460.40      -.560      .287714
           bits  9 to 16  2503.57       .050      .520109
           bits 10 to 17  2500.53       .007      .502989
           bits 11 to 18  2551.73       .732      .767771
           bits 12 to 19  2413.97     -1.217      .111859
           bits 13 to 20  2539.52       .559      .711874
           bits 14 to 21  2445.73      -.767      .221401
           bits 15 to 22  2664.64      2.328      .990054
           bits 16 to 23  2523.30       .330      .629128
           bits 17 to 24  2459.51      -.573      .283460
           bits 18 to 25  2410.58     -1.265      .103009
           bits 19 to 26  2498.92      -.015      .493910
           bits 20 to 27  2494.79      -.074      .470634
           bits 21 to 28  2504.90       .069      .527640
           bits 22 to 29  2337.90     -2.292      .010940
           bits 23 to 30  2486.08      -.197      .421963
           bits 24 to 31  2445.35      -.773      .219815
           bits 25 to 32  2524.27       .343      .634262

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

     :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
     ::               THIS IS A PARKING LOT TEST                      ::
     :: In a square of side 100, randomly "park" a car---a circle of  ::
     :: radius 1.   Then try to park a 2nd, a 3rd, and so on, each    ::
     :: time parking "by ear".  That is, if an attempt to park a car  ::
     :: causes a crash with one already parked, try again at a new    ::
     :: random location. (To avoid path problems, consider parking    ::
     :: helicopters rather than cars.)   Each attempt leads to either ::
```

```
:: a crash or a success, the latter followed by an increment to  ::
:: the list of cars already parked. If we plot n:  the number of ::
:: attempts, versus k::  the number successfully parked, we get a::
:: curve that should be similar to those provided by a perfect   ::
:: random number generator.  Theory for the behavior of such a   ::
:: random curve seems beyond reach, and as graphics displays are ::
:: not available for this battery of tests, a simple characteriz ::
:: ation of the random experiment is used: k, the number of cars ::
:: successfully parked after n=12,000 attempts. Simulation shows ::
:: that k should average 3523 with sigma 21.9 and is very close  ::
:: to normally distributed.  Thus (k-3523)/21.9 should be a st-  ::
:: andard normal variable, which, converted to a uniform varia-  ::
:: ble, provides input to a KSTEST based on a sample of 10.      ::
:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
      CDPARK: result of ten tests on file block1.rng
       Of 12,000 tries, the average no. of successes
            should be 3523 with sigma=21.9
       Successes: 3529    z-score:   .274 p-value: .607947
       Successes: 3535    z-score:   .548 p-value: .708135
       Successes: 3527    z-score:   .183 p-value: .572463
       Successes: 3524    z-score:   .046 p-value: .518210
       Successes: 3524    z-score:   .046 p-value: .518210
       Successes: 3517    z-score:  -.274 p-value: .392053
       Successes: 3536    z-score:   .594 p-value: .723613
       Successes: 3505    z-score:  -.822 p-value: .205562
       Successes: 3519    z-score:  -.183 p-value: .427537
       Successes: 3516    z-score:  -.320 p-value: .374623

      square size   avg. no.  parked   sample sigma
        100.            3523.200        8.908
       KSTEST for the above 10: p=  .744364

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
::                THE MINIMUM DISTANCE TEST                      ::
:: It does this 100 times::   choose n=8000 random points in a   ::
:: square of side 10000.  Find d, the minimum distance between   ::
:: the (n^2-n)/2 pairs of points.  If the points are truly inde- ::
:: pendent uniform, then d^2, the square of the minimum distance ::
:: should be (very close to) exponentially distributed with mean ::
:: .995 .  Thus 1-exp(-d^2/.995) should be uniform on [0,1) and  ::
:: a KSTEST on the resulting 100 values serves as a test of uni- ::
:: formity for random points in the square. Test numbers=0 mod 5 ::
:: are printed but the KSTEST is based on the full set of 100    ::
:: random choices of 8000 points in the 10000x10000 square.      ::
:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
          This is the MINIMUM DISTANCE test
         for random integers in the file block1.rng
Sample no.    d^2     avg     equiv uni
      5    1.0065   1.5492     .636364
     10     .2319   1.1350     .207893
     15     .0331    .9154     .032708
     20    1.6184   1.0570     .803390
     25    4.1402   1.2587     .984408
     30    1.6313   1.2451     .805924
     35    1.2023   1.1596     .701304
```

```
          40    3.4760   1.1402     .969604
          45     .6647   1.0950     .487296
          50     .3223   1.1371     .276722
          55     .9412   1.0731     .611666
          60    1.4237   1.0313     .760894
          65    1.5687   1.0156     .793326
          70     .6165   1.0044     .461818
          75    2.1718    .9816     .887268
          80     .0906    .9964     .086994
          85    1.9667   1.0413     .861458
          90    1.4027   1.0183     .755804
          95     .5892    .9915     .446853
         100     .7267    .9896     .518271
     MINIMUM DISTANCE TEST for block1.rng
          Result of KS test on 20 transformed mindist^2's:
                                  p-value= .236388

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

     :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
     ::               THE 3DSPHERES TEST                              ::
     :: Choose  4000 random points in a cube of edge 1000.  At each   ::
     :: point, center a sphere large enough to reach the next closest ::
     :: point. Then the volume of the smallest such sphere is (very   ::
     :: close to) exponentially distributed with mean 120pi/3.  Thus  ::
     :: the radius cubed is exponential with mean 30. (The mean is    ::
     :: obtained by extensive simulation).  The 3DSPHERES test gener- ::
     :: ates 4000 such spheres 20 times.  Each min radius cubed leads ::
     :: to a uniform variable by means of 1-exp(-r^3/30.), then a     ::
     ::  KSTEST is done on the 20 p-values.                           ::
     :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
               The 3DSPHERES test for file block1.rng
 sample no:  1     r^3=  14.966     p-value= .39278
 sample no:  2     r^3=  43.029     p-value= .76172
 sample no:  3     r^3=   8.329     p-value= .24242
 sample no:  4     r^3=  45.484     p-value= .78044
 sample no:  5     r^3=  23.237     p-value= .53910
 sample no:  6     r^3=   1.065     p-value= .03486
 sample no:  7     r^3=   5.846     p-value= .17706
 sample no:  8     r^3=  67.995     p-value= .89632
 sample no:  9     r^3=  51.001     p-value= .81732
 sample no: 10     r^3=  55.741     p-value= .84402
 sample no: 11     r^3=   4.885     p-value= .15028
 sample no: 12     r^3=  25.221     p-value= .56859
 sample no: 13     r^3=  54.727     p-value= .83866
 sample no: 14     r^3=   3.484     p-value= .10965
 sample no: 15     r^3=  25.865     p-value= .57776
 sample no: 16     r^3=   1.602     p-value= .05200
 sample no: 17     r^3=  32.831     p-value= .66525
 sample no: 18     r^3=   2.263     p-value= .07265
 sample no: 19     r^3=  40.078     p-value= .73709
 sample no: 20     r^3=  43.991     p-value= .76924
  A KS test is applied to those 20 p-values.
---------------------------------------------------------
       3DSPHERES test for file block1.rng           p-value= .414786
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
```

```
::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
::      This is the SQEEZE test                                 ::
::  Random integers are floated to get uniforms on [0,1). Start- ::
::  ing with k=2^31=2147483647, the test finds j, the number of  ::
::  iterations necessary to reduce k to 1, using the reduction   ::
::  k=ceiling(k*U), with U provided by floating integers from    ::
::  the file being tested.  Such j's are found 100,000 times,    ::
::  then counts for the number of times j was <=6,7,...,47,>=48  ::
::  are used to provide a chi-square test for cell frequencies.  ::
::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
       RESULTS OF SQUEEZE TEST FOR block1.rng
    Table of standardized frequency counts
( (obs-exp)/sqrt(exp) )^2
   for j taking values <=6,7,8,...,47,>=48:
 -.1      .5     1.1     -.3     -.1      .4
 2.1      .3     -.4      .7      .6      .1
 -.5      .7    -2.1     1.9     -.6      .7
 -.2     -.3      .3     -.2     1.0    -1.7
  .1     -.6     -.8      .1    -2.1      .9
  .0     -.5     2.1     1.2      .6    -1.7
 -.7      .5    -1.2     1.0      .9    -1.0
  .8
      Chi-square with 42 degrees of freedom: 42.412
         z-score=   .045  p-value= .546769
```

```
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
::             The  OVERLAPPING SUMS test                       ::
:: Integers are floated to get a sequence U(1),U(2),... of uni-  ::
:: form [0,1) variables.  Then overlapping sums,                ::
::   S(1)=U(1)+...+U(100), S2=U(2)+...+U(101),... are formed.   ::
:: The S's are virtually normal with a certain covariance mat-  ::
:: rix.  A linear transformation of the S's converts them to a  ::
:: sequence of independent standard normals, which are converted ::
:: to uniform variables for a KSTEST. The  p-values from ten    ::
:: KSTESTs are given still another KSTEST.                      ::
::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
           Test no.  1      p-value  .185986
           Test no.  2      p-value  .704747
           Test no.  3      p-value  .338745
           Test no.  4      p-value  .919336
           Test no.  5      p-value  .820863
           Test no.  6      p-value  .600849
           Test no.  7      p-value  .973676
           Test no.  8      p-value  .549063
           Test no.  9      p-value  .335171
           Test no. 10      p-value  .705374
   Results of the OSUM test for block1.rng
        KSTEST on the above 10 p-values:  .602188

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
::     This is the RUNS test.  It counts runs up, and runs down, ::
:: in a sequence of uniform [0,1) variables, obtained by float-  ::
```

```
:: ing the 32-bit integers in the specified file. This example   ::
:: shows how runs are counted:  .123,.357,.789,.425,.224,.416,.95::
:: contains an up-run of length 3, a down-run of length 2 and an ::
:: up-run of (at least) 2, depending on the next values.  The    ::
:: covariance matrices for the runs-up and runs-down are well    ::
:: known, leading to chisquare tests for quadratic forms in the  ::
:: weak inverses of the covariance matrices.  Runs are counted   ::
:: for sequences of length 10,000.  This is done ten times. Then ::
:: repeated.                                                     ::
:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
      The RUNS test for file block1.rng
Up and down runs in a sample of 10000
```

```
                 Run test for block1.rng     :
           runs up; ks test for 10 p's: .006170
         runs down; ks test for 10 p's: .286105
                 Run test for block1.rng     :
           runs up; ks test for 10 p's: .118531
         runs down; ks test for 10 p's: .287039

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

     :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
     :: This is the CRAPS TEST. It plays 200,000 games of craps, finds::
     :: the number of wins and the number of throws necessary to end  ::
     :: each game.  The number of wins should be (very close to) a    ::
     :: normal with mean 200000p and variance 200000p(1-p), with      ::
     :: p=244/495.  Throws necessary to complete the game can vary    ::
     :: from 1 to infinity, but counts for all>21 are lumped with 21. ::
     :: A chi-square test is made on the no.-of-throws cell counts.   ::
     :: Each 32-bit integer from the test file provides the value for ::
     :: the throw of a die, by floating to [0,1), multiplying by 6    ::
     :: and taking 1 plus the integer part of the result.             ::
     :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
                Results of craps test for block1.rng
  No. of wins:  Observed Expected
                                98704    98585.86
                  98704= No. of wins, z-score=  .528 pvalue= .70139
   Analysis of Throws-per-Game:
 Chisq=  24.22 for 20 degrees of freedom, p=  .76713
               Throws Observed Expected  Chisq     Sum
                  1    66272    66666.7   2.336    2.336
                  2    37680    37654.3    .018    2.354
                  3    27082    26954.7    .601    2.955
                  4    19479    19313.5   1.419    4.374
                  5    13915    13851.4    .292    4.666
                  6    10046     9943.5   1.056    5.721
                  7     7228     7145.0    .964    6.685
                  8     5064     5139.1   1.097    7.782
                  9     3705     3699.9    .007    7.789
                 10     2713     2666.3    .818    8.607
                 11     1901     1923.3    .259    8.866
                 12     1428     1388.7   1.110    9.976
                 13      921     1003.7   6.816   16.792
                 14      724      726.1    .006   16.799
                 15      523      525.8    .015   16.814
                 16      334      381.2   5.833   22.647
```

```
                 17      262      276.5    .764   23.411
                 18      202      200.8    .007   23.418
                 19      142      146.0    .109   23.527
                 20      106      106.2    .000   23.527
                 21      273      287.1    .694   24.221
            SUMMARY  FOR block1.rng
                p-value for no. of wins: .701387
                p-value for throws/game: .767132

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

 Results of DIEHARD battery of tests sent to file report1.txt




      NOTE: Most of the tests in DIEHARD return a p-value, which
      should be uniform on [0,1) if the input file contains truly
      independent random bits.   Those p-values are obtained by
      p=F(X), where F is the assumed distribution of the sample
      random variable X---often normal. But that assumed F is just
      an asymptotic approximation, for which the fit will be worst
      in the tails. Thus you should not be surprised with
      occasional p-values near 0 or 1, such as .0012 or .9983.
      When a bit stream really FAILS BIG, you will get p's of 0 or
      1 to six or more places.  By all means, do not, as a
      Statistician might, think that a p < .025 or p> .975 means
      that the RNG has "failed the test at the .05 level".  Such
      p's happen among the hundreds that DIEHARD produces, even
      with good RNG's.  So keep in mind that " p happens".
    :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
    ::            This is the BIRTHDAY SPACINGS TEST                 ::
    :: Choose m birthdays in a year of n days.  List the spacings    ::
    :: between the birthdays.  If j is the number of values that     ::
    :: occur more than once in that list, then j is asymptotically   ::
    :: Poisson distributed with mean m^3/(4n).  Experience shows n   ::
    :: must be quite large, say n>=2^18, for comparing the results   ::
    :: to the Poisson distribution with that mean.  This test uses   ::
    :: n=2^24 and m=2^9,  so that the underlying distribution for j  ::
    :: is taken to be Poisson with lambda=2^27/(2^26)=2.  A sample   ::
    :: of 500 j's is taken, and a chi-square goodness of fit test    ::
    :: provides a p value.  The first test uses bits 1-24 (counting  ::
    :: from the left) from integers in the specified file.           ::
    ::   Then the file is closed and reopened. Next, bits 2-25 are   ::
    :: used to provide birthdays, then 3-26 and so on to bits 9-32.  ::
    :: Each set of bits provides a p-value, and the nine p-values    ::
    :: provide a sample for a KSTEST.                                ::
    :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
 BIRTHDAY SPACINGS TEST, M= 512 N=2**24 LAMBDA=  2.0000
           Results for block2.rng
                   For a sample of size 500:     mean
           block2.rng      using bits  1 to 24   1.962
  duplicate       number       number
  spacings       observed     expected
        0          83.       67.668
        1         119.      135.335
        2         122.      135.335
```

```
      3          110.       90.224
      4           45.       45.112
      5           18.       18.045
 6 to INF          3.        8.282
Chisquare with  6 d.o.f. =    14.46 p-value=  .975134
 :::::::::::::::::::::::::::::::::::::::::
                 For a sample of size 500:     mean
         block2.rng      using bits  2 to 25   2.002
 duplicate       number       number
 spacings       observed     expected
      0           62.       67.668
      1          137.      135.335
      2          140.      135.335
      3           90.       90.224
      4           49.       45.112
      5           15.       18.045
 6 to INF          7.        8.282
Chisquare with  6 d.o.f. =     1.70 p-value=  .055171
 :::::::::::::::::::::::::::::::::::::::::
                 For a sample of size 500:     mean
         block2.rng      using bits  3 to 26   1.848
 duplicate       number       number
 spacings       observed     expected
      0           66.       67.668
      1          172.      135.335
      2          121.      135.335
      3           90.       90.224
      4           26.       45.112
      5           17.       18.045
 6 to INF          8.        8.282
Chisquare with  6 d.o.f. =    19.66 p-value=  .996817
 :::::::::::::::::::::::::::::::::::::::::
                 For a sample of size 500:     mean
         block2.rng      using bits  4 to 27   1.980
 duplicate       number       number
 spacings       observed     expected
      0           64.       67.668
      1          140.      135.335
      2          152.      135.335
      3           69.       90.224
      4           44.       45.112
      5           24.       18.045
 6 to INF          7.        8.282
Chisquare with  6 d.o.f. =     9.60 p-value=  .857237
 :::::::::::::::::::::::::::::::::::::::::
                 For a sample of size 500:     mean
         block2.rng      using bits  5 to 28   1.930
 duplicate       number       number
 spacings       observed     expected
      0           79.       67.668
      1          139.      135.335
      2          125.      135.335
      3           83.       90.224
      4           50.       45.112
      5           18.       18.045
 6 to INF          6.        8.282
Chisquare with  6 d.o.f. =     4.52 p-value=  .393750
```

```
 :::::::::::::::::::::::::::::::::::::::::
                 For a sample of size 500:     mean
         block2.rng      using bits  6 to 29   1.980
 duplicate       number       number
 spacings       observed     expected
      0          73.       67.668
      1         131.      135.335
      2         136.      135.335
      3          90.       90.224
      4          40.       45.112
      5          23.       18.045
 6 to INF         7.        8.282
Chisquare with  6 d.o.f. =     2.70 p-value=  .154706
 :::::::::::::::::::::::::::::::::::::::::
                 For a sample of size 500:     mean
         block2.rng      using bits  7 to 30   1.994
 duplicate       number       number
 spacings       observed     expected
      0          69.       67.668
      1         140.      135.335
      2         123.      135.335
      3          97.       90.224
      4          45.       45.112
      5          16.       18.045
 6 to INF        10.        8.282
Chisquare with  6 d.o.f. =     2.41 p-value=  .121461
 :::::::::::::::::::::::::::::::::::::::::
                 For a sample of size 500:     mean
         block2.rng      using bits  8 to 31   1.932
 duplicate       number       number
 spacings       observed     expected
      0          68.       67.668
      1         148.      135.335
      2         131.      135.335
      3          85.       90.224
      4          48.       45.112
      5          14.       18.045
 6 to INF         6.        8.282
Chisquare with  6 d.o.f. =     3.35 p-value=  .235971
 :::::::::::::::::::::::::::::::::::::::::
                 For a sample of size 500:     mean
         block2.rng      using bits  9 to 32   2.070
 duplicate       number       number
 spacings       observed     expected
      0          54.       67.668
      1         126.      135.335
      2         146.      135.335
      3         109.       90.224
      4          43.       45.112
      5          16.       18.045
 6 to INF         6.        8.282
Chisquare with  6 d.o.f. =     9.11 p-value=  .832611
 :::::::::::::::::::::::::::::::::::::::::
  The 9 p-values were
       .975134   .055171   .996817   .857237   .393750
       .154706   .121461   .235971   .832611
 A KSTEST for the 9 p-values yields  .782829
```

```
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

     :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
     ::             THE OVERLAPPING 5-PERMUTATION TEST                ::
     :: This is the OPERM5 test.  It looks at a sequence of one mill- ::
     :: ion 32-bit random integers.  Each set of five consecutive     ::
     :: integers can be in one of 120 states, for the 5! possible or- ::
     :: derings of five numbers.  Thus the 5th, 6th, 7th,...numbers   ::
     :: each provide a state. As many thousands of state transitions  ::
     :: are observed,  cumulative counts are made of the number of    ::
     :: occurences of each state.  Then the quadratic form in the     ::
     :: weak inverse of the 120x120 covariance matrix yields a test   ::
     :: equivalent to the likelihood ratio test that the 120 cell     ::
     :: counts came from the specified (asymptotically) normal dis-   ::
     :: tribution with the specified 120x120 covariance matrix (with  ::
     :: rank 99).  This version uses 1,000,000 integers, twice.       ::
     :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
           OPERM5 test for file block2.rng
     For a sample of 1,000,000 consecutive 5-tuples,
 chisquare for 99 degrees of freedom=107.290; p-value= .732627
           OPERM5 test for file block2.rng
     For a sample of 1,000,000 consecutive 5-tuples,
 chisquare for 99 degrees of freedom= 98.253; p-value= .497661
     :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
     :: This is the BINARY RANK TEST for 31x31 matrices. The leftmost ::
     :: 31 bits of 31 random integers from the test sequence are used ::
     :: to form a 31x31 binary matrix over the field {0,1}. The rank  ::
     :: is determined. That rank can be from 0 to 31, but ranks< 28   ::
     :: are rare, and their counts are pooled with those for rank 28. ::
     :: Ranks are found for 40,000 such random matrices and a chisqua-::
     :: re test is performed on counts for ranks 31,30,29 and <=28.   ::
     :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
    Binary rank test for block2.rng
         Rank test for 31x31 binary matrices:
        rows from leftmost 31 bits of each 32-bit integer
      rank   observed  expected (o-e)^2/e  sum
        28       209     211.4   .027655     .028
        29      5139    5134.0   .004850     .033
        30     23230   23103.0   .697618     .730
        31     11422   11551.5  1.452326    2.182
  chisquare= 2.182 for 3 d. of f.; p-value= .534383
--------------------------------------------------------------
     :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
     :: This is the BINARY RANK TEST for 32x32 matrices. A random 32x ::
     :: 32 binary matrix is formed, each row a 32-bit random integer. ::
     :: The rank is determined. That rank can be from 0 to 32, ranks  ::
     :: less than 29 are rare, and their counts are pooled with those ::
     :: for rank 29.  Ranks are found for 40,000 such random matrices ::
     :: and a chisquare test is performed on counts for ranks  32,31, ::
     :: 30 and <=29.                                                  ::
     :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
    Binary rank test for block2.rng
         Rank test for 32x32 binary matrices:
        rows from leftmost 32 bits of each 32-bit integer
      rank   observed  expected (o-e)^2/e  sum
        29       206     211.4   .138848     .139
```

```
       30      4984    5134.0  4.383138    4.522
       31     23211   23103.0   .504430    5.026
       32     11599   11551.5   .195120    5.222
  chisquare= 5.222 for 3 d. of f.; p-value= .854879
--------------------------------------------------------------

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
     :: This is the BINARY RANK TEST for 6x8 matrices.  From each of  ::
     :: six random 32-bit integers from the generator under test, a   ::
     :: specified byte is chosen, and the resulting six bytes form a  ::
     :: 6x8 binary matrix whose rank is determined.  That rank can be ::
     :: from 0 to 6, but ranks 0,1,2,3 are rare; their counts are     ::
     :: pooled with those for rank 4. Ranks are found for 100,000     ::
     :: random matrices, and a chi-square test is performed on        ::
     :: counts for ranks 6,5 and <=4.                                 ::
     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
         Binary Rank Test for block2.rng
        Rank of a 6x8 binary matrix,
     rows formed from eight bits of the RNG block2.rng
     b-rank test for bits  1 to  8
                     OBSERVED   EXPECTED     (O-E)^2/E      SUM
          r<=4          967       944.3         .546        .546
          r =5        21627     21743.9         .628       1.174
          r =6        77406     77311.8         .115       1.289
                        p=1-exp(-SUM/2)= .47504
        Rank of a 6x8 binary matrix,
     rows formed from eight bits of the RNG block2.rng
     b-rank test for bits  2 to  9
                     OBSERVED   EXPECTED     (O-E)^2/E      SUM
          r<=4          972       944.3         .812        .812
          r =5        21723     21743.9         .020        .833
          r =6        77305     77311.8         .001        .833
                        p=1-exp(-SUM/2)= .34070
        Rank of a 6x8 binary matrix,
     rows formed from eight bits of the RNG block2.rng
     b-rank test for bits  3 to 10
                     OBSERVED   EXPECTED     (O-E)^2/E      SUM
          r<=4          949       944.3         .023        .023
          r =5        21883     21743.9         .890        .913
          r =6        77168     77311.8         .267       1.181
                        p=1-exp(-SUM/2)= .44587
        Rank of a 6x8 binary matrix,
     rows formed from eight bits of the RNG block2.rng
     b-rank test for bits  4 to 11
                     OBSERVED   EXPECTED     (O-E)^2/E      SUM
          r<=4          962       944.3         .332        .332
          r =5        21636     21743.9         .535        .867
          r =6        77402     77311.8         .105        .972
                        p=1-exp(-SUM/2)= .38504
        Rank of a 6x8 binary matrix,
     rows formed from eight bits of the RNG block2.rng
     b-rank test for bits  5 to 12
                     OBSERVED   EXPECTED     (O-E)^2/E      SUM
          r<=4          965       944.3         .454        .454
          r =5        21704     21743.9         .073        .527
```

```
     r =6        77331     77311.8        .005         .532
                  p=1-exp(-SUM/2)= .23344
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block2.rng
b-rank test for bits  6 to 13
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          921       944.3        .575         .575
     r =5        22059     21743.9       4.566        5.141
     r =6        77020     77311.8       1.101        6.243
                  p=1-exp(-SUM/2)= .95590
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block2.rng
b-rank test for bits  7 to 14
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          995       944.3       2.722        2.722
     r =5        21779     21743.9        .057        2.779
     r =6        77226     77311.8        .095        2.874
                  p=1-exp(-SUM/2)= .76234
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block2.rng
b-rank test for bits  8 to 15
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          949       944.3        .023         .023
     r =5        21865     21743.9        .674         .698
     r =6        77186     77311.8        .205         .903
                  p=1-exp(-SUM/2)= .36318
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block2.rng
b-rank test for bits  9 to 16
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          943       944.3        .002         .002
     r =5        21766     21743.9        .022         .024
     r =6        77291     77311.8        .006         .030
                  p=1-exp(-SUM/2)= .01482
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block2.rng
b-rank test for bits 10 to 17
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          942       944.3        .006         .006
     r =5        21734     21743.9        .005         .010
     r =6        77324     77311.8        .002         .012
                  p=1-exp(-SUM/2)= .00600
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block2.rng
b-rank test for bits 11 to 18
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          966       944.3        .499         .499
     r =5        21637     21743.9        .526        1.024
     r =6        77397     77311.8        .094        1.118
                  p=1-exp(-SUM/2)= .42823
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block2.rng
b-rank test for bits 12 to 19
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          919       944.3        .678         .678
     r =5        21884     21743.9        .903        1.581
     r =6        77197     77311.8        .170        1.751
```

```
                  p=1-exp(-SUM/2)= .58336
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block2.rng
b-rank test for bits 13 to 20
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          931       944.3        .187         .187
     r =5        21680     21743.9        .188         .375
     r =6        77389     77311.8        .077         .452
                  p=1-exp(-SUM/2)= .20237
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block2.rng
b-rank test for bits 14 to 21
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          999       944.3       3.168        3.168
     r =5        21913     21743.9       1.315        4.483
     r =6        77088     77311.8        .648        5.131
                  p=1-exp(-SUM/2)= .92313
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block2.rng
b-rank test for bits 15 to 22
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          974       944.3        .934         .934
     r =5        21909     21743.9       1.254        2.188
     r =6        77117     77311.8        .491        2.678
                  p=1-exp(-SUM/2)= .73796
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block2.rng
b-rank test for bits 16 to 23
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          979       944.3       1.275        1.275
     r =5        21730     21743.9        .009        1.284
     r =6        77291     77311.8        .006        1.290
                  p=1-exp(-SUM/2)= .47521
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block2.rng
b-rank test for bits 17 to 24
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          941       944.3        .012         .012
     r =5        21869     21743.9        .720         .731
     r =6        77190     77311.8        .192         .923
                  p=1-exp(-SUM/2)= .36972
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block2.rng
b-rank test for bits 18 to 25
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          934       944.3        .112         .112
     r =5        21797     21743.9        .130         .242
     r =6        77269     77311.8        .024         .266
                  p=1-exp(-SUM/2)= .12442
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block2.rng
b-rank test for bits 19 to 26
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4         1004       944.3       3.774        3.774
     r =5        21852     21743.9        .537        4.312
     r =6        77144     77311.8        .364        4.676
                  p=1-exp(-SUM/2)= .90347
```

```
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block2.rng
b-rank test for bits 20 to 27
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          904       944.3       1.720       1.720
     r =5        21822     21743.9        .281       2.001
     r =6        77274     77311.8        .018       2.019
                   p=1-exp(-SUM/2)= .63560
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block2.rng
b-rank test for bits 21 to 28
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          919       944.3        .678        .678
     r =5        21652     21743.9        .388       1.066
     r =6        77429     77311.8        .178       1.244
                   p=1-exp(-SUM/2)= .46313
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block2.rng
b-rank test for bits 22 to 29
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          944       944.3        .000        .000
     r =5        21661     21743.9        .316        .316
     r =6        77395     77311.8        .090        .406
                   p=1-exp(-SUM/2)= .18359
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block2.rng
b-rank test for bits 23 to 30
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          983       944.3       1.586       1.586
     r =5        21732     21743.9        .007       1.592
     r =6        77285     77311.8        .009       1.602
                   p=1-exp(-SUM/2)= .55106
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block2.rng
b-rank test for bits 24 to 31
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          947       944.3        .008        .008
     r =5        21491     21743.9       2.941       2.949
     r =6        77562     77311.8        .810       3.759
                   p=1-exp(-SUM/2)= .84732
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block2.rng
b-rank test for bits 25 to 32
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          925       944.3        .395        .395
     r =5        21771     21743.9        .034        .428
     r =6        77304     77311.8        .001        .429
                   p=1-exp(-SUM/2)= .19309
  TEST SUMMARY, 25 tests on 100,000 random 6x8 matrices
These should be 25 uniform [0,1] random variables:
    .475042     .340703     .445870     .385036     .233443
    .955900     .762344     .363182     .014816     .006002
    .428233     .583363     .202373     .923134     .737955
    .475207     .369720     .124424     .903469     .635598
    .463126     .183594     .551060     .847322     .193086
  brank test summary for block2.rng
     The KS test for those 25 supposed UNI's yields
```

```
                    KS p-value= .308795

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
     ::                   THE BITSTREAM TEST                         ::
     :: The file under test is viewed as a stream of bits. Call them  ::
     :: b1,b2,... .  Consider an alphabet with two "letters", 0 and 1 ::
     :: and think of the stream of bits as a succession of 20-letter  ::
     :: "words", overlapping.  Thus the first word is b1b2...b20, the ::
     :: second is b2b3...b21, and so on.  The bitstream test counts   ::
     :: the number of missing 20-letter (20-bit) words in a string of ::
     :: 2^21 overlapping 20-letter words.  There are 2^20 possible 20 ::
     :: letter words.  For a truly random string of 2^21+19 bits, the ::
     :: number of missing words j should be (very close to) normally  ::
     :: distributed with mean 141,909 and sigma 428.  Thus            ::
     ::  (j-141909)/428 should be a standard normal variate (z score) ::
     :: that leads to a uniform [0,1) p value.  The test is repeated  ::
     :: twenty times.                                                ::
     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
 THE OVERLAPPING 20-tuples BITSTREAM  TEST, 20 BITS PER WORD, N words
   This test uses N=2^21 and samples the bitstream 20 times.
  No. missing words should average  141909. with sigma=428.
---------------------------------------------------------
 tst no  1:  142089 missing words,     .42 sigmas from mean, p-value= .66268
 tst no  2:  141658 missing words,    -.59 sigmas from mean, p-value= .27853
 tst no  3:  141207 missing words,   -1.64 sigmas from mean, p-value= .05040
 tst no  4:  141804 missing words,    -.25 sigmas from mean, p-value= .40280
 tst no  5:  141515 missing words,    -.92 sigmas from mean, p-value= .17844
 tst no  6:  142354 missing words,    1.04 sigmas from mean, p-value= .85059
 tst no  7:  141985 missing words,     .18 sigmas from mean, p-value= .57017
 tst no  8:  142084 missing words,     .41 sigmas from mean, p-value= .65840
 tst no  9:  142174 missing words,     .62 sigmas from mean, p-value= .73184
 tst no 10:  141404 missing words,   -1.18 sigmas from mean, p-value= .11887
 tst no 11:  142315 missing words,     .95 sigmas from mean, p-value= .82839
 tst no 12:  141954 missing words,     .10 sigmas from mean, p-value= .54156
 tst no 13:  142301 missing words,     .92 sigmas from mean, p-value= .81994
 tst no 14:  141976 missing words,     .16 sigmas from mean, p-value= .56189
 tst no 15:  141667 missing words,    -.57 sigmas from mean, p-value= .28563
 tst no 16:  141557 missing words,    -.82 sigmas from mean, p-value= .20520
 tst no 17:  141884 missing words,    -.06 sigmas from mean, p-value= .47641
 tst no 18:  141038 missing words,   -2.04 sigmas from mean, p-value= .02088
 tst no 19:  141555 missing words,    -.83 sigmas from mean, p-value= .20387
 tst no 20:  142210 missing words,     .70 sigmas from mean, p-value= .75882

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
     ::             The tests OPSO, OQSO and DNA                     ::
     ::         OPSO means Overlapping-Pairs-Sparse-Occupancy        ::
     :: The OPSO test considers 2-letter words from an alphabet of   ::
     :: 1024 letters.  Each letter is determined by a specified ten  ::
     :: bits from a 32-bit integer in the sequence to be tested. OPSO ::
     :: generates  2^21 (overlapping) 2-letter words  (from 2^21+1   ::
     :: "keystrokes")  and counts the number of missing words---that  ::
     :: is 2-letter words which do not appear in the entire sequence. ::
     :: That count should be very close to normally distributed with  ::
```

```
    :: mean 141,909, sigma 290. Thus (missingwrds-141909)/290 should ::
    :: be a standard normal variable. The OPSO test takes 32 bits at ::
    :: a time from the test file and uses a designated set of ten    ::
    :: consecutive bits. It then restarts the file for the next de-  ::
    :: signated 10 bits, and so on.                                  ::
    ::                                                               ::
    ::     OQSO means Overlapping-Quadruples-Sparse-Occupancy        ::
    ::   The test OQSO is similar, except that it considers 4-letter ::
    :: words from an alphabet of 32 letters, each letter determined  ::
    :: by a designated string of 5 consecutive bits from the test    ::
    :: file, elements of which are assumed 32-bit random integers.   ::
    :: The mean number of missing words in a sequence of 2^21 four-  ::
    :: letter words,  (2^21+3 "keystrokes"), is again 141909, with   ::
    :: sigma = 295.  The mean is based on theory; sigma comes from   ::
    :: extensive simulation.                                         ::
    ::                                                               ::
    ::    The DNA test considers an alphabet of 4 letters::  C,G,A,T,::
    :: determined by two designated bits in the sequence of random   ::
    :: integers being tested.  It considers 10-letter words, so that ::
    :: as in OPSO and OQSO, there are 2^20 possible words, and the   ::
    :: mean number of missing words from a string of 2^21  (over-    ::
    :: lapping)  10-letter  words (2^21+9 "keystrokes") is 141909.   ::
    :: The standard deviation sigma=339 was determined as for OQSO   ::
    :: by simulation.  (Sigma for OPSO, 290, is the true value (to   ::
    :: three places), not determined by simulation.                  ::
    :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
 OPSO test for generator block2.rng
  Output: No. missing words (mw), equiv normal variate (z), p-value (p)
                                                           mw     z     p
    OPSO for block2.rng      using bits 23 to 32        141926   .057  .5229
    OPSO for block2.rng      using bits 22 to 31        141655  -.877  .1902
    OPSO for block2.rng      using bits 21 to 30        141540 -1.274  .1014
    OPSO for block2.rng      using bits 20 to 29        141766  -.494  .3106
    OPSO for block2.rng      using bits 19 to 28        141791  -.408  .3416
    OPSO for block2.rng      using bits 18 to 27        141599 -1.070  .1423
    OPSO for block2.rng      using bits 17 to 26        141693  -.746  .2278
    OPSO for block2.rng      using bits 16 to 25        142020   .382  .6486
    OPSO for block2.rng      using bits 15 to 24        141910   .002  .5009
    OPSO for block2.rng      using bits 14 to 23        142107   .682  .7523
    OPSO for block2.rng      using bits 13 to 22        141345 -1.946  .0258
    OPSO for block2.rng      using bits 12 to 21        141982   .251  .5989
    OPSO for block2.rng      using bits 11 to 20        142194   .982  .8369
    OPSO for block2.rng      using bits 10 to 19        141871  -.132  .4474
    OPSO for block2.rng      using bits  9 to 18        142285  1.295  .9024
    OPSO for block2.rng      using bits  8 to 17        142064   .533  .7031
    OPSO for block2.rng      using bits  7 to 16        142174   .913  .8193
    OPSO for block2.rng      using bits  6 to 15        141620  -.998  .1592
    OPSO for block2.rng      using bits  5 to 14        142040   .451  .6739
    OPSO for block2.rng      using bits  4 to 13        141643  -.918  .1792
    OPSO for block2.rng      using bits  3 to 12        141938   .099  .5394
    OPSO for block2.rng      using bits  2 to 11        142019   .378  .6474
    OPSO for block2.rng      using bits  1 to 10        142118   .720  .7641
 OQSO test for generator block2.rng
  Output: No. missing words (mw), equiv normal variate (z), p-value (p)
                                                           mw     z     p
    OQSO for block2.rng      using bits 28 to 32        141417 -1.669  .0476
    OQSO for block2.rng      using bits 27 to 31        142525  2.087  .9816
```

```
   OQSO for block2.rng      using bits 26 to 30        141540 -1.252  .1053
   OQSO for block2.rng      using bits 25 to 29        142191   .955  .8302
   OQSO for block2.rng      using bits 24 to 28        141787  -.415  .3392
   OQSO for block2.rng      using bits 23 to 27        142081   .582  .7197
   OQSO for block2.rng      using bits 22 to 26        141832  -.262  .3966
   OQSO for block2.rng      using bits 21 to 25        141808  -.343  .3656
   OQSO for block2.rng      using bits 20 to 24        142083   .589  .7220
   OQSO for block2.rng      using bits 19 to 23        141832  -.262  .3966
   OQSO for block2.rng      using bits 18 to 22        142059   .507  .6940
   OQSO for block2.rng      using bits 17 to 21        141768  -.479  .3159
   OQSO for block2.rng      using bits 16 to 20        141728  -.615  .2694
   OQSO for block2.rng      using bits 15 to 19        141843  -.225  .4111
   OQSO for block2.rng      using bits 14 to 18        141865  -.150  .4403
   OQSO for block2.rng      using bits 13 to 17        142128   .741  .7707
   OQSO for block2.rng      using bits 12 to 16        142486  1.955  .9747
   OQSO for block2.rng      using bits 11 to 15        142250  1.155  .8759
   OQSO for block2.rng      using bits 10 to 14        141631  -.943  .1727
   OQSO for block2.rng      using bits  9 to 13        142457  1.857  .9683
   OQSO for block2.rng      using bits  8 to 12        142031   .412  .6600
   OQSO for block2.rng      using bits  7 to 11        141823  -.293  .3849
   OQSO for block2.rng      using bits  6 to 10        142458  1.860  .9686
   OQSO for block2.rng      using bits  5 to  9        141521 -1.316  .0940
   OQSO for block2.rng      using bits  4 to  8        141551 -1.215  .1122
   OQSO for block2.rng      using bits  3 to  7        141940   .104  .5414
   OQSO for block2.rng      using bits  2 to  6        141965   .189  .5748
   OQSO for block2.rng      using bits  1 to  5        141535 -1.269  .1022
 DNA test for generator block2.rng
 Output: No. missing words (mw), equiv normal variate (z), p-value (p)
                                                          mw     z     p
    DNA for block2.rng      using bits 31 to 32        141866  -.128  .4491
    DNA for block2.rng      using bits 30 to 31        142219   .913  .8195
    DNA for block2.rng      using bits 29 to 30        141918   .026  .5102
    DNA for block2.rng      using bits 28 to 29        142359  1.326  .9077
    DNA for block2.rng      using bits 27 to 28        141892  -.051  .4796
    DNA for block2.rng      using bits 26 to 27        142418  1.501  .9333
    DNA for block2.rng      using bits 25 to 26        142116   .610  .7290
    DNA for block2.rng      using bits 24 to 25        141643  -.786  .2160
    DNA for block2.rng      using bits 23 to 24        141628  -.830  .2033
    DNA for block2.rng      using bits 22 to 23        141877  -.095  .4620
    DNA for block2.rng      using bits 21 to 22        141905  -.013  .4949
    DNA for block2.rng      using bits 20 to 21        142216   .905  .8172
    DNA for block2.rng      using bits 19 to 20        141803  -.314  .3769
    DNA for block2.rng      using bits 18 to 19        141599  -.915  .1800
    DNA for block2.rng      using bits 17 to 18        141462 -1.320  .0935
    DNA for block2.rng      using bits 16 to 17        142067   .465  .6791
    DNA for block2.rng      using bits 15 to 16        142715  2.377  .9913
    DNA for block2.rng      using bits 14 to 15        142352  1.306  .9042
    DNA for block2.rng      using bits 13 to 14        142220   .916  .8203
    DNA for block2.rng      using bits 12 to 13        141977   .200  .5791
    DNA for block2.rng      using bits 11 to 12        142416  1.495  .9325
    DNA for block2.rng      using bits 10 to 11        142532  1.837  .9669
    DNA for block2.rng      using bits  9 to 10        141785  -.367  .3569
    DNA for block2.rng      using bits  8 to  9        142192   .834  .7978
    DNA for block2.rng      using bits  7 to  8        141734  -.517  .3025
    DNA for block2.rng      using bits  6 to  7        141865  -.131  .4480
    DNA for block2.rng      using bits  5 to  6        141496 -1.219  .1114
    DNA for block2.rng      using bits  4 to  5        142407  1.468  .9290
```

```
    DNA for block2.rng      using bits  3 to  4        141585  -.957  .1694
    DNA for block2.rng      using bits  2 to  3        142239   .972  .8346
    DNA for block2.rng      using bits  1 to  2        142094   .545  .7070

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
     ::     This is the COUNT-THE-1's TEST on a stream of bytes.      ::
     :: Consider the file under test as a stream of bytes (four per   ::
     :: 32 bit integer).  Each byte can contain from 0 to 8 1's,      ::
     :: with probabilities 1,8,28,56,70,56,28,8,1 over 256.  Now let  ::
     :: the stream of bytes provide a string of overlapping  5-letter ::
     :: words, each "letter" taking values A,B,C,D,E. The letters are ::
     :: determined by the number of 1's in a byte::  0,1,or 2 yield A,::
     :: 3 yields B, 4 yields C, 5 yields D and 6,7 or 8 yield E. Thus ::
     :: we have a monkey at a typewriter hitting five keys with vari- ::
     :: ous probabilities (37,56,70,56,37 over 256).  There are 5^5   ::
     :: possible 5-letter words, and from a string of 256,000 (over-  ::
     :: lapping) 5-letter words, counts are made on the frequencies   ::
     :: for each word.   The quadratic form in the weak inverse of    ::
     :: the covariance matrix of the cell counts provides a chisquare ::
     :: test::  Q5-Q4, the difference of the naive Pearson sums of    ::
     :: (OBS-EXP)^2/EXP on counts for 5- and 4-letter cell counts.    ::
     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
   Test results for block2.rng
 Chi-square with 5^5-5^4=2500 d.of f. for sample size:2560000
                             chisquare  equiv normal  p-value
  Results fo COUNT-THE-1's in successive bytes:
 byte stream for block2.rng       2541.92       .593      .723350
 byte stream for block2.rng       2464.41      -.503      .307360

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
     ::     This is the COUNT-THE-1's TEST for specific bytes.        ::
     :: Consider the file under test as a stream of 32-bit integers.  ::
     :: From each integer, a specific byte is chosen , say the left-  ::
     :: most::  bits 1 to 8. Each byte can contain from 0 to 8 1's,   ::
     :: with probabilitie 1,8,28,56,70,56,28,8,1 over 256.  Now let   ::
     :: the specified bytes from successive integers provide a string ::
     :: of (overlapping) 5-letter words, each "letter" taking values  ::
     :: A,B,C,D,E. The letters are determined  by the number of 1's,  ::
     :: in that byte::  0,1,or 2 ---> A, 3 ---> B, 4 ---> C, 5 ---> D,::
     :: and  6,7 or 8 ---> E.  Thus we have a monkey at a typewriter  ::
     :: hitting five keys with with various probabilities::  37,56,70,::
     :: 56,37 over 256. There are 5^5 possible 5-letter words, and    ::
     :: from a string of 256,000 (overlapping) 5-letter words, counts ::
     :: are made on the frequencies for each word. The quadratic form ::
     :: in the weak inverse of the covariance matrix of the cell      ::
     :: counts provides a chisquare test::  Q5-Q4, the difference of  ::
     :: the naive Pearson  sums of (OBS-EXP)^2/EXP on counts for 5-   ::
     :: and 4-letter cell counts.                                     ::
     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
 Chi-square with 5^5-5^4=2500 d.of f. for sample size: 256000
                      chisquare  equiv normal  p value
  Results for COUNT-THE-1's in specified bytes:
           bits  1 to  8  2506.72       .095      .537841
```

```
bits  2 to  9  2521.51       .304      .619530
bits  3 to 10  2565.24       .923      .821889
bits  4 to 11  2513.45       .190      .575437
bits  5 to 12  2604.65      1.480      .930567
bits  6 to 13  2495.09      -.069      .472321
bits  7 to 14  2468.40      -.447      .327491
bits  8 to 15  2497.45      -.036      .485615
bits  9 to 16  2495.38      -.065      .473977
bits 10 to 17  2606.28      1.503      .933579
bits 11 to 18  2539.53       .559      .711939
bits 12 to 19  2449.31      -.717      .236707
bits 13 to 20  2469.84      -.427      .334864
bits 14 to 21  2456.07      -.621      .267209
bits 15 to 22  2415.45     -1.196      .115904
bits 16 to 23  2433.21      -.945      .172431
bits 17 to 24  2422.24     -1.100      .135726
bits 18 to 25  2506.90       .098      .538885
bits 19 to 26  2421.74     -1.107      .134193
bits 20 to 27  2514.15       .200      .579280
bits 21 to 28  2471.12      -.408      .341483
bits 22 to 29  2506.29       .089      .535446
bits 23 to 30  2516.45       .233      .591973
bits 24 to 31  2577.72      1.099      .864157
bits 25 to 32  2422.23     -1.100      .135690
```

```
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

      ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
      ::                THIS IS A PARKING LOT TEST                      ::
      :: In a square of side 100, randomly "park" a car---a circle of  ::
      :: radius 1.   Then try to park a 2nd, a 3rd, and so on, each    ::
      :: time parking "by ear".  That is, if an attempt to park a car  ::
      :: causes a crash with one already parked, try again at a new    ::
      :: random location. (To avoid path problems, consider parking    ::
      :: helicopters rather than cars.)   Each attempt leads to either ::
      :: a crash or a success, the latter followed by an increment to  ::
      :: the list of cars already parked. If we plot n:  the number of ::
      :: attempts, versus k::  the number successfully parked, we get a::
      :: curve that should be similar to those provided by a perfect   ::
      :: random number generator.  Theory for the behavior of such a   ::
      :: random curve seems beyond reach, and as graphics displays are ::
      :: not available for this battery of tests, a simple characteriz ::
      :: ation of the random experiment is used: k, the number of cars ::
      :: successfully parked after n=12,000 attempts. Simulation shows ::
      :: that k should average 3523 with sigma 21.9 and is very close  ::
      :: to normally distributed.  Thus (k-3523)/21.9 should be a st-  ::
      :: andard normal variable, which, converted to a uniform varia-  ::
      :: ble, provides input to a KSTEST based on a sample of 10.      ::
      ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
            CDPARK: result of ten tests on file block2.rng
             Of 12,000 tries, the average no. of successes
                  should be 3523 with sigma=21.9
             Successes: 3551    z-score:  1.279 p-value: .899470
             Successes: 3513    z-score:  -.457 p-value: .323972
             Successes: 3519    z-score:  -.183 p-value: .427537
             Successes: 3504    z-score:  -.868 p-value: .192812
             Successes: 3580    z-score:  2.603 p-value: .995376
```

```
    Successes: 3518    z-score:  -.228 p-value: .409702
    Successes: 3542    z-score:   .868 p-value: .807188
    Successes: 3549    z-score:  1.187 p-value: .882429
    Successes: 3568    z-score:  2.055 p-value: .980051
    Successes: 3509    z-score:  -.639 p-value: .261324

   square size   avg. no.  parked   sample sigma
     100.            3535.300       25.066
    KSTEST for the above 10: p=  .876667

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
     ::               THE MINIMUM DISTANCE TEST                         ::
     :: It does this 100 times::   choose n=8000 random points in a   ::
     :: square of side 10000.  Find d, the minimum distance between   ::
     :: the (n^2-n)/2 pairs of points.  If the points are truly inde- ::
     :: pendent uniform, then d^2, the square of the minimum distance ::
     :: should be (very close to) exponentially distributed with mean ::
     :: .995 .  Thus 1-exp(-d^2/.995) should be uniform on [0,1) and  ::
     :: a KSTEST on the resulting 100 values serves as a test of uni- ::
     :: formity for random points in the square. Test numbers=0 mod 5 ::
     :: are printed but the KSTEST is based on the full set of 100    ::
     :: random choices of 8000 points in the 10000x10000 square.      ::
     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
               This is the MINIMUM DISTANCE test
              for random integers in the file block2.rng
     Sample no.    d^2     avg     equiv uni
           5     .3650    .4842     .307079
          10    1.1963    .7874     .699504
          15    2.3154   1.0809     .902413
          20    1.0490   1.0153     .651555
          25    1.8590   1.0995     .845628
          30    4.3542   1.1730     .987425
          35    1.0572   1.1370     .654400
          40     .4340   1.1454     .353528
          45     .1654   1.1251     .153181
          50    2.1226   1.2228     .881545
          55     .5501   1.2954     .424682
          60     .7003   1.2717     .505286
          65     .4166   1.2288     .342112
          70     .8006   1.3118     .552747
          75     .3872   1.2541     .322394
          80     .6747   1.2308     .492421
          85     .5588   1.1974     .429705
          90     .0874   1.1699     .084066
          95     .4232   1.1198     .346417
         100    2.1494   1.1011     .884697
     MINIMUM DISTANCE TEST for block2.rng
          Result of KS test on 20 transformed mindist^2's:
                                  p-value= .516688

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
     ::              THE 3DSPHERES TEST                                 ::
     :: Choose  4000 random points in a cube of edge 1000.  At each   ::
```

```
     :: point, center a sphere large enough to reach the next closest ::
     :: point. Then the volume of the smallest such sphere is (very   ::
     :: close to) exponentially distributed with mean 120pi/3.  Thus  ::
     :: the radius cubed is exponential with mean 30. (The mean is    ::
     :: obtained by extensive simulation).  The 3DSPHERES test gener- ::
     :: ates 4000 such spheres 20 times.  Each min radius cubed leads ::
     :: to a uniform variable by means of 1-exp(-r^3/30.), then a     ::
     ::  KSTEST is done on the 20 p-values.                           ::
     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
              The 3DSPHERES test for file block2.rng
 sample no:  1     r^3= 212.550     p-value= .99916
 sample no:  2     r^3=  31.020     p-value= .64442
 sample no:  3     r^3=  24.176     p-value= .55330
 sample no:  4     r^3=  15.143     p-value= .39635
 sample no:  5     r^3=    .072     p-value= .00238
 sample no:  6     r^3=  59.281     p-value= .86138
 sample no:  7     r^3=  20.472     p-value= .49460
 sample no:  8     r^3=  42.123     p-value= .75441
 sample no:  9     r^3=  14.192     p-value= .37690
 sample no: 10     r^3=    .534     p-value= .01765
 sample no: 11     r^3=  15.351     p-value= .40052
 sample no: 12     r^3=  25.875     p-value= .57790
 sample no: 13     r^3=   1.325     p-value= .04320
 sample no: 14     r^3=  14.000     p-value= .37292
 sample no: 15     r^3=  81.065     p-value= .93294
 sample no: 16     r^3=  27.442     p-value= .59937
 sample no: 17     r^3=  61.888     p-value= .87292
 sample no: 18     r^3=  23.766     p-value= .54715
 sample no: 19     r^3=  14.383     p-value= .38087
 sample no: 20     r^3=  13.988     p-value= .37266
  A KS test is applied to those 20 p-values.
---------------------------------------------------------
      3DSPHERES test for file block2.rng           p-value= .578757
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
     ::      This is the SQEEZE test                                  ::
     ::  Random integers are floated to get uniforms on [0,1). Start- ::
     ::  ing with k=2^31=2147483647, the test finds j, the number of  ::
     ::  iterations necessary to reduce k to 1, using the reduction   ::
     ::  k=ceiling(k*U), with U provided by floating integers from    ::
     ::  the file being tested.  Such j's are found 100,000 times,    ::
     ::  then counts for the number of times j was <=6,7,...,47,>=48  ::
     ::  are used to provide a chi-square test for cell frequencies.  ::
     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
            RESULTS OF SQUEEZE TEST FOR block2.rng
         Table of standardized frequency counts
     ( (obs-exp)/sqrt(exp) )^2
        for j taking values <=6,7,8,...,47,>=48:
     -.8     -.3     1.1      .5     -.1      .3
    -1.6      .2     1.2      .8      .3    -1.3
     1.0      .0     -.7      .0      .5      .3
     -.2      .7      .3     -.5      .3      .2
      .0    -1.2    -1.5      .9    -1.3     -.3
    -1.6      .8     1.8    -2.3     -.5      .4
     -.7    -1.0     3.0    -1.8     1.6    -1.0
     -.1
```

```
          Chi-square with 42 degrees of freedom: 47.267
             z-score=   .575  p-value= .733980
```

```
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

     :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
     ::             The  OVERLAPPING SUMS test                      ::
     :: Integers are floated to get a sequence U(1),U(2),... of uni-  ::
     :: form [0,1) variables.  Then overlapping sums,                 ::
     ::   S(1)=U(1)+...+U(100), S2=U(2)+...+U(101),... are formed.    ::
     :: The S's are virtually normal with a certain covariance mat-   ::
     :: rix.  A linear transformation of the S's converts them to a   ::
     :: sequence of independent standard normals, which are converted ::
     :: to uniform variables for a KSTEST. The  p-values from ten     ::
     :: KSTESTs are given still another KSTEST.                       ::
     :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
                Test no.  1      p-value  .784409
                Test no.  2      p-value  .397363
                Test no.  3      p-value  .716635
                Test no.  4      p-value  .911424
                Test no.  5      p-value  .858380
                Test no.  6      p-value  .840614
                Test no.  7      p-value  .700915
                Test no.  8      p-value  .261768
                Test no.  9      p-value  .683462
                Test no. 10      p-value  .108448
   Results of the OSUM test for block2.rng
        KSTEST on the above 10 p-values:  .775137

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

     :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
     ::     This is the RUNS test.  It counts runs up, and runs down, ::
     :: in a sequence of uniform [0,1) variables, obtained by float-  ::
     :: ing the 32-bit integers in the specified file. This example   ::
     :: shows how runs are counted:  .123,.357,.789,.425,.224,.416,.95::
     :: contains an up-run of length 3, a down-run of length 2 and an ::
     :: up-run of (at least) 2, depending on the next values.  The    ::
     :: covariance matrices for the runs-up and runs-down are well    ::
     :: known, leading to chisquare tests for quadratic forms in the  ::
     :: weak inverses of the covariance matrices.  Runs are counted   ::
     :: for sequences of length 10,000.  This is done ten times. Then ::
     :: repeated.                                                     ::
     :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
          The RUNS test for file block2.rng
     Up and down runs in a sample of 10000
```

```
                 Run test for block2.rng     :
       runs up; ks test for 10 p's: .430820
     runs down; ks test for 10 p's: .632125
                 Run test for block2.rng     :
       runs up; ks test for 10 p's: .684980
     runs down; ks test for 10 p's: .443339

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
```

```
:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
:: This is the CRAPS TEST. It plays 200,000 games of craps, finds::
:: the number of wins and the number of throws necessary to end  ::
:: each game.  The number of wins should be (very close to) a    ::
:: normal with mean 200000p and variance 200000p(1-p), with      ::
:: p=244/495.  Throws necessary to complete the game can vary    ::
:: from 1 to infinity, but counts for all>21 are lumped with 21. ::
:: A chi-square test is made on the no.-of-throws cell counts.   ::
:: Each 32-bit integer from the test file provides the value for ::
:: the throw of a die, by floating to [0,1), multiplying by 6    ::
:: and taking 1 plus the integer part of the result.             ::
:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
           Results of craps test for block2.rng
 No. of wins:  Observed Expected
                               98409    98585.86
             98409= No. of wins, z-score= -.791 pvalue= .21447
  Analysis of Throws-per-Game:
Chisq=  15.60 for 20 degrees of freedom, p=  .25885
          Throws Observed Expected  Chisq     Sum
             1    66322    66666.7   1.782    1.782
             2    37778    37654.3    .406    2.188
             3    27082    26954.7    .601    2.789
             4    19315    19313.5    .000    2.789
             5    14051    13851.4   2.876    5.665
             6     9854     9943.5    .806    6.471
             7     7135     7145.0    .014    6.485
             8     5058     5139.1   1.279    7.764
             9     3663     3699.9    .367    8.132
            10     2745     2666.3   2.323   10.455
            11     1912     1923.3    .067   10.521
            12     1429     1388.7   1.167   11.689
            13     1010     1003.7    .039   11.728
            14      704      726.1    .675   12.403
            15      508      525.8    .605   13.008
            16      399      381.2    .836   13.844
            17      286      276.5    .324   14.168
            18      193      200.8    .305   14.473
            19      158      146.0    .989   15.462
            20      110      106.2    .135   15.597
            21      288      287.1    .003   15.599
        SUMMARY  FOR block2.rng
           p-value for no. of wins: .214467
           p-value for throws/game: .258854

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

 Results of DIEHARD battery of tests sent to file report2.txt
```

NOTE: Most of the tests in DIEHARD return a p-value, which should be uniform on [0,1) if the input file contains truly independent random bits. Those p-values are obtained by p=F(X), where F is the assumed distribution of the sample random variable X---often normal. But that assumed F is just an asymptotic approximation, for which the fit will be worst

```
      in the tails. Thus you should not be surprised with
      occasional p-values near 0 or 1, such as .0012 or .9983.
      When a bit stream really FAILS BIG, you will get p's of 0 or
      1 to six or more places.  By all means, do not, as a
      Statistician might, think that a p < .025 or p> .975 means
      that the RNG has "failed the test at the .05 level".  Such
      p's happen among the hundreds that DIEHARD produces, even
      with good RNG's.  So keep in mind that " p happens".
    :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
    ::            This is the BIRTHDAY SPACINGS TEST                 ::
    :: Choose m birthdays in a year of n days.  List the spacings    ::
    :: between the birthdays.  If j is the number of values that     ::
    :: occur more than once in that list, then j is asymptotically   ::
    :: Poisson distributed with mean m^3/(4n).  Experience shows n   ::
    :: must be quite large, say n>=2^18, for comparing the results   ::
    :: to the Poisson distribution with that mean.  This test uses   ::
    :: n=2^24 and m=2^9,  so that the underlying distribution for j  ::
    :: is taken to be Poisson with lambda=2^27/(2^26)=2.  A sample   ::
    :: of 500 j's is taken, and a chi-square goodness of fit test    ::
    :: provides a p value.  The first test uses bits 1-24 (counting  ::
    :: from the left) from integers in the specified file.           ::
    ::   Then the file is closed and reopened. Next, bits 2-25 are   ::
    :: used to provide birthdays, then 3-26 and so on to bits 9-32.  ::
    :: Each set of bits provides a p-value, and the nine p-values    ::
    :: provide a sample for a KSTEST.                                ::
    :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
 BIRTHDAY SPACINGS TEST, M= 512 N=2**24 LAMBDA=  2.0000
           Results for block3.rng
                   For a sample of size 500:     mean
           block3.rng      using bits  1 to 24   2.042
  duplicate       number       number
  spacings       observed     expected
        0          61.       67.668
        1         141.      135.335
        2         132.      135.335
        3          96.       90.224
        4          37.       45.112
        5          21.       18.045
  6 to INF         12.        8.282
 Chisquare with  6 d.o.f. =     4.96 p-value=  .450798
  :::::::::::::::::::::::::::::::::::::::::
                   For a sample of size 500:     mean
           block3.rng      using bits  2 to 25   2.074
  duplicate       number       number
  spacings       observed     expected
        0          69.       67.668
        1         126.      135.335
        2         123.      135.335
        3          96.       90.224
        4          60.       45.112
        5          22.       18.045
  6 to INF          4.        8.282
 Chisquare with  6 d.o.f. =    10.16 p-value=  .881871
  :::::::::::::::::::::::::::::::::::::::::
                   For a sample of size 500:     mean
           block3.rng      using bits  3 to 26   1.974
  duplicate       number       number
```

```
 spacings       observed     expected
      0           63.       67.668
      1          154.      135.335
      2          130.      135.335
      3           81.       90.224
      4           43.       45.112
      5           19.       18.045
 6 to INF         10.        8.282
Chisquare with  6 d.o.f. =     4.56 p-value=  .398019
 :::::::::::::::::::::::::::::::::::::::::
                  For a sample of size 500:     mean
         block3.rng      using bits  4 to 27   1.966
 duplicate       number       number
 spacings       observed     expected
      0           63.       67.668
      1          150.      135.335
      2          132.      135.335
      3           88.       90.224
      4           45.       45.112
      5           10.       18.045
 6 to INF         12.        8.282
Chisquare with  6 d.o.f. =     7.30 p-value=  .706362
 :::::::::::::::::::::::::::::::::::::::::
                  For a sample of size 500:     mean
         block3.rng      using bits  5 to 28   2.016
 duplicate       number       number
 spacings       observed     expected
      0           56.       67.668
      1          147.      135.335
      2          135.      135.335
      3           97.       90.224
      4           40.       45.112
      5           16.       18.045
 6 to INF          9.        8.282
Chisquare with  6 d.o.f. =     4.40 p-value=  .377313
 :::::::::::::::::::::::::::::::::::::::::
                  For a sample of size 500:     mean
         block3.rng      using bits  6 to 29   1.994
 duplicate       number       number
 spacings       observed     expected
      0           72.       67.668
      1          129.      135.335
      2          133.      135.335
      3           91.       90.224
      4           53.       45.112
      5           15.       18.045
 6 to INF          7.        8.282
Chisquare with  6 d.o.f. =     2.71 p-value=  .156018
 :::::::::::::::::::::::::::::::::::::::::
                  For a sample of size 500:     mean
         block3.rng      using bits  7 to 30   2.062
 duplicate       number       number
 spacings       observed     expected
      0           62.       67.668
      1          132.      135.335
      2          132.      135.335
      3          100.       90.224
```

```
       4          49.       45.112
       5          15.       18.045
  6 to INF        10.        8.282
 Chisquare with  6 d.o.f. =     2.90 p-value=  .179179
  :::::::::::::::::::::::::::::::::::::::::
                  For a sample of size 500:     mean
          block3.rng      using bits  8 to 31   1.982
  duplicate       number       number
  spacings       observed     expected
       0          67.       67.668
       1         139.      135.335
       2         135.      135.335
       3          90.       90.224
       4          41.       45.112
       5          22.       18.045
  6 to INF         6.        8.282
 Chisquare with  6 d.o.f. =     1.98 p-value=  .078257
  :::::::::::::::::::::::::::::::::::::::::
                  For a sample of size 500:     mean
          block3.rng      using bits  9 to 32   1.968
  duplicate       number       number
  spacings       observed     expected
       0          66.       67.668
       1         145.      135.335
       2         133.      135.335
       3          91.       90.224
       4          39.       45.112
       5          18.       18.045
  6 to INF         8.        8.282
 Chisquare with  6 d.o.f. =     1.62 p-value=  .048579
  :::::::::::::::::::::::::::::::::::::::::
   The 9 p-values were
        .450798   .881871   .398019   .706362   .377313
        .156018   .179179   .078257   .048579
  A KSTEST for the 9 p-values yields  .727333

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
     ::             THE OVERLAPPING 5-PERMUTATION TEST               ::
     :: This is the OPERM5 test.  It looks at a sequence of one mill- ::
     :: ion 32-bit random integers.  Each set of five consecutive     ::
     :: integers can be in one of 120 states, for the 5! possible or- ::
     :: derings of five numbers.  Thus the 5th, 6th, 7th,...numbers   ::
     :: each provide a state. As many thousands of state transitions  ::
     :: are observed,  cumulative counts are made of the number of    ::
     :: occurences of each state.  Then the quadratic form in the     ::
     :: weak inverse of the 120x120 covariance matrix yields a test   ::
     :: equivalent to the likelihood ratio test that the 120 cell     ::
     :: counts came from the specified (asymptotically) normal dis-   ::
     :: tribution with the specified 120x120 covariance matrix (with  ::
     :: rank 99).  This version uses 1,000,000 integers, twice.       ::
     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
           OPERM5 test for file block3.rng
     For a sample of 1,000,000 consecutive 5-tuples,
 chisquare for 99 degrees of freedom= 96.461; p-value= .446464
           OPERM5 test for file block3.rng
```

```
     For a sample of 1,000,000 consecutive 5-tuples,
 chisquare for 99 degrees of freedom=123.505; p-value= .951760
     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
     :: This is the BINARY RANK TEST for 31x31 matrices. The leftmost ::
     :: 31 bits of 31 random integers from the test sequence are used ::
     :: to form a 31x31 binary matrix over the field {0,1}. The rank  ::
     :: is determined. That rank can be from 0 to 31, but ranks< 28   ::
     :: are rare, and their counts are pooled with those for rank 28. ::
     :: Ranks are found for 40,000 such random matrices and a chisqua-::
     :: re test is performed on counts for ranks 31,30,29 and <=28.   ::
     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
    Binary rank test for block3.rng
         Rank test for 31x31 binary matrices:
        rows from leftmost 31 bits of each 32-bit integer
      rank   observed  expected (o-e)^2/e  sum
        28       178     211.4  5.282254    5.282
        29      5228    5134.0  1.720696    7.003
        30     22946   23103.0  1.067553    8.071
        31     11648   11551.5   .805741    8.876
  chisquare= 8.876 for 3 d. of f.; p-value= .970462
--------------------------------------------------------------
     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
     :: This is the BINARY RANK TEST for 32x32 matrices. A random 32x ::
     :: 32 binary matrix is formed, each row a 32-bit random integer. ::
     :: The rank is determined. That rank can be from 0 to 32, ranks  ::
     :: less than 29 are rare, and their counts are pooled with those ::
     :: for rank 29.  Ranks are found for 40,000 such random matrices ::
     :: and a chisquare test is performed on counts for ranks  32,31, ::
     :: 30 and <=29.                                                  ::
     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
    Binary rank test for block3.rng
         Rank test for 32x32 binary matrices:
        rows from leftmost 32 bits of each 32-bit integer
      rank   observed  expected (o-e)^2/e  sum
        29       268     211.4 15.143090   15.143
        30      5076    5134.0   .655470   15.799
        31     23102   23103.0   .000047   15.799
        32     11554   11551.5   .000531   15.799
  chisquare=15.799 for 3 d. of f.; p-value= .998788
--------------------------------------------------------------

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
     :: This is the BINARY RANK TEST for 6x8 matrices.  From each of  ::
     :: six random 32-bit integers from the generator under test, a   ::
     :: specified byte is chosen, and the resulting six bytes form a  ::
     :: 6x8 binary matrix whose rank is determined.  That rank can be ::
     :: from 0 to 6, but ranks 0,1,2,3 are rare; their counts are     ::
     :: pooled with those for rank 4. Ranks are found for 100,000     ::
     :: random matrices, and a chi-square test is performed on       ::
     :: counts for ranks 6,5 and <=4.                                 ::
     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
         Binary Rank Test for block3.rng
        Rank of a 6x8 binary matrix,
     rows formed from eight bits of the RNG block3.rng
     b-rank test for bits  1 to  8
```

```
              OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4        948       944.3        .014         .014
     r =5      21954     21743.9       2.030        2.045
     r =6      77098     77311.8        .591        2.636
                 p=1-exp(-SUM/2)= .73231
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block3.rng
b-rank test for bits  2 to  9
              OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4        970       944.3        .699         .699
     r =5      21850     21743.9        .518        1.217
     r =6      77180     77311.8        .225        1.442
                 p=1-exp(-SUM/2)= .51369
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block3.rng
b-rank test for bits  3 to 10
              OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4        922       944.3        .527         .527
     r =5      21758     21743.9        .009         .536
     r =6      77320     77311.8        .001         .537
                 p=1-exp(-SUM/2)= .23536
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block3.rng
b-rank test for bits  4 to 11
              OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4        996       944.3       2.830        2.830
     r =5      21700     21743.9        .089        2.919
     r =6      77304     77311.8        .001        2.920
                 p=1-exp(-SUM/2)= .76774
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block3.rng
b-rank test for bits  5 to 12
              OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4        951       944.3        .048         .048
     r =5      21744     21743.9        .000         .048
     r =6      77305     77311.8        .001         .048
                 p=1-exp(-SUM/2)= .02377
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block3.rng
b-rank test for bits  6 to 13
              OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4        943       944.3        .002         .002
     r =5      21700     21743.9        .089         .090
     r =6      77357     77311.8        .026         .117
                 p=1-exp(-SUM/2)= .05675
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block3.rng
b-rank test for bits  7 to 14
              OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4        958       944.3        .199         .199
     r =5      21785     21743.9        .078         .276
     r =6      77257     77311.8        .039         .315
                 p=1-exp(-SUM/2)= .14583
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block3.rng
b-rank test for bits  8 to 15
              OBSERVED   EXPECTED     (O-E)^2/E      SUM
```

```
     r<=4          942       944.3          .006          .006
     r =5        21770     21743.9          .031          .037
     r =6        77288     77311.8          .007          .044
                    p=1-exp(-SUM/2)= .02189
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block3.rng
b-rank test for bits  9 to 16
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          921       944.3          .575          .575
     r =5        21910     21743.9         1.269         1.844
     r =6        77169     77311.8          .264         2.108
                    p=1-exp(-SUM/2)= .65138
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block3.rng
b-rank test for bits 10 to 17
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          961       944.3          .295          .295
     r =5        21777     21743.9          .050          .346
     r =6        77262     77311.8          .032          .378
                    p=1-exp(-SUM/2)= .17212
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block3.rng
b-rank test for bits 11 to 18
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          914       944.3          .972          .972
     r =5        21844     21743.9          .461         1.433
     r =6        77242     77311.8          .063         1.496
                    p=1-exp(-SUM/2)= .52673
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block3.rng
b-rank test for bits 12 to 19
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          938       944.3          .042          .042
     r =5        21919     21743.9         1.410         1.452
     r =6        77143     77311.8          .369         1.821
                    p=1-exp(-SUM/2)= .59761
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block3.rng
b-rank test for bits 13 to 20
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          955       944.3          .121          .121
     r =5        22095     21743.9         5.669         5.790
     r =6        76950     77311.8         1.693         7.484
                    p=1-exp(-SUM/2)= .97629
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block3.rng
b-rank test for bits 14 to 21
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          972       944.3          .812          .812
     r =5        21674     21743.9          .225         1.037
     r =6        77354     77311.8          .023         1.060
                    p=1-exp(-SUM/2)= .41146
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block3.rng
b-rank test for bits 15 to 22
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          947       944.3          .008          .008
```

```
     r =5        21690     21743.9        .134        .141
     r =6        77363     77311.8        .034        .175
                  p=1-exp(-SUM/2)= .08388
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block3.rng
b-rank test for bits 16 to 23
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          910       944.3       1.246       1.246
     r =5        22038     21743.9       3.978       5.224
     r =6        77052     77311.8        .873       6.097
                  p=1-exp(-SUM/2)= .95257
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block3.rng
b-rank test for bits 17 to 24
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          907       944.3       1.473       1.473
     r =5        21945     21743.9       1.860       3.333
     r =6        77148     77311.8        .347       3.680
                  p=1-exp(-SUM/2)= .84121
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block3.rng
b-rank test for bits 18 to 25
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          947       944.3        .008        .008
     r =5        21578     21743.9       1.266       1.273
     r =6        77475     77311.8        .344       1.618
                  p=1-exp(-SUM/2)= .55469
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block3.rng
b-rank test for bits 19 to 26
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          975       944.3        .998        .998
     r =5        21720     21743.9        .026       1.024
     r =6        77305     77311.8        .001       1.025
                  p=1-exp(-SUM/2)= .40096
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block3.rng
b-rank test for bits 20 to 27
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          993       944.3       2.511       2.511
     r =5        21946     21743.9       1.878       4.390
     r =6        77061     77311.8        .814       5.204
                  p=1-exp(-SUM/2)= .92586
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block3.rng
b-rank test for bits 21 to 28
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          975       944.3        .998        .998
     r =5        21836     21743.9        .390       1.388
     r =6        77189     77311.8        .195       1.583
                  p=1-exp(-SUM/2)= .54687
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block3.rng
b-rank test for bits 22 to 29
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          986       944.3       1.841       1.841
     r =5        21867     21743.9        .697       2.538
```

```
        r =6         77147     77311.8        .351       2.890
                       p=1-exp(-SUM/2)= .76420
      Rank of a 6x8 binary matrix,
   rows formed from eight bits of the RNG block3.rng
   b-rank test for bits 23 to 30
                   OBSERVED   EXPECTED     (O-E)^2/E      SUM
        r<=4           951       944.3        .048        .048
        r =5         21797     21743.9        .130        .177
        r =6         77252     77311.8        .046        .223
                       p=1-exp(-SUM/2)= .10571
      Rank of a 6x8 binary matrix,
   rows formed from eight bits of the RNG block3.rng
   b-rank test for bits 24 to 31
                   OBSERVED   EXPECTED     (O-E)^2/E      SUM
        r<=4           940       944.3        .020        .020
        r =5         21777     21743.9        .050        .070
        r =6         77283     77311.8        .011        .081
                       p=1-exp(-SUM/2)= .03955
      Rank of a 6x8 binary matrix,
   rows formed from eight bits of the RNG block3.rng
   b-rank test for bits 25 to 32
                   OBSERVED   EXPECTED     (O-E)^2/E      SUM
        r<=4           942       944.3        .006        .006
        r =5         21773     21743.9        .039        .045
        r =6         77285     77311.8        .009        .054
                       p=1-exp(-SUM/2)= .02656
  TEST SUMMARY, 25 tests on 100,000 random 6x8 matrices
 These should be 25 uniform [0,1] random variables:
    .732309     .513686     .235357     .767744     .023772
    .056750     .145834     .021890     .651385     .172116
    .526727     .597610     .976289     .411458     .083885
    .952568     .841214     .554691     .400965     .925856
    .546874     .764203     .105711     .039552     .026563
  brank test summary for block3.rng
      The KS test for those 25 supposed UNI's yields
                  KS p-value= .805665

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

    :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
    ::                   THE BITSTREAM TEST                          ::
    :: The file under test is viewed as a stream of bits. Call them  ::
    :: b1,b2,... .  Consider an alphabet with two "letters", 0 and 1 ::
    :: and think of the stream of bits as a succession of 20-letter  ::
    :: "words", overlapping.  Thus the first word is b1b2...b20, the ::
    :: second is b2b3...b21, and so on.  The bitstream test counts   ::
    :: the number of missing 20-letter (20-bit) words in a string of ::
    :: 2^21 overlapping 20-letter words.  There are 2^20 possible 20 ::
    :: letter words.  For a truly random string of 2^21+19 bits, the ::
    :: number of missing words j should be (very close to) normally  ::
    :: distributed with mean 141,909 and sigma 428.  Thus            ::
    ::  (j-141909)/428 should be a standard normal variate (z score) ::
    :: that leads to a uniform [0,1) p value.  The test is repeated  ::
    :: twenty times.                                                 ::
    :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
 THE OVERLAPPING 20-tuples BITSTREAM  TEST, 20 BITS PER WORD, N words
   This test uses N=2^21 and samples the bitstream 20 times.
```

```
  No. missing words should average  141909. with sigma=428.
---------------------------------------------------------
 tst no  1:  141543 missing words,    -.86 sigmas from mean, p-value= .19602
 tst no  2:  142352 missing words,    1.03 sigmas from mean, p-value= .84950
 tst no  3:  141698 missing words,    -.49 sigmas from mean, p-value= .31074
 tst no  4:  141445 missing words,   -1.08 sigmas from mean, p-value= .13899
 tst no  5:  142411 missing words,    1.17 sigmas from mean, p-value= .87943
 tst no  6:  142308 missing words,     .93 sigmas from mean, p-value= .82420
 tst no  7:  141282 missing words,   -1.47 sigmas from mean, p-value= .07136
 tst no  8:  142059 missing words,     .35 sigmas from mean, p-value= .63672
 tst no  9:  141818 missing words,    -.21 sigmas from mean, p-value= .41551
 tst no 10:  141039 missing words,   -2.03 sigmas from mean, p-value= .02100
 tst no 11:  141849 missing words,    -.14 sigmas from mean, p-value= .44395
 tst no 12:  141853 missing words,    -.13 sigmas from mean, p-value= .44765
 tst no 13:  141787 missing words,    -.29 sigmas from mean, p-value= .38751
 tst no 14:  141854 missing words,    -.13 sigmas from mean, p-value= .44857
 tst no 15:  142041 missing words,     .31 sigmas from mean, p-value= .62082
 tst no 16:  142027 missing words,     .27 sigmas from mean, p-value= .60832
 tst no 17:  142106 missing words,     .46 sigmas from mean, p-value= .67707
 tst no 18:  141772 missing words,    -.32 sigmas from mean, p-value= .37416
 tst no 19:  142356 missing words,    1.04 sigmas from mean, p-value= .85167
 tst no 20:  141874 missing words,    -.08 sigmas from mean, p-value= .46711

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
     ::             The tests OPSO, OQSO and DNA                     ::
     ::         OPSO means Overlapping-Pairs-Sparse-Occupancy        ::
     :: The OPSO test considers 2-letter words from an alphabet of    ::
     :: 1024 letters.  Each letter is determined by a specified ten   ::
     :: bits from a 32-bit integer in the sequence to be tested. OPSO ::
     :: generates  2^21 (overlapping) 2-letter words  (from 2^21+1    ::
     :: "keystrokes")  and counts the number of missing words---that  ::
     :: is 2-letter words which do not appear in the entire sequence. ::
     :: That count should be very close to normally distributed with  ::
     :: mean 141,909, sigma 290. Thus (missingwrds-141909)/290 should ::
     :: be a standard normal variable. The OPSO test takes 32 bits at ::
     :: a time from the test file and uses a designated set of ten    ::
     :: consecutive bits. It then restarts the file for the next de-  ::
     :: signated 10 bits, and so on.                                  ::
     ::                                                               ::
     ::     OQSO means Overlapping-Quadruples-Sparse-Occupancy        ::
     ::   The test OQSO is similar, except that it considers 4-letter ::
     :: words from an alphabet of 32 letters, each letter determined  ::
     :: by a designated string of 5 consecutive bits from the test    ::
     :: file, elements of which are assumed 32-bit random integers.   ::
     :: The mean number of missing words in a sequence of 2^21 four-  ::
     :: letter words,  (2^21+3 "keystrokes"), is again 141909, with   ::
     :: sigma = 295.  The mean is based on theory; sigma comes from   ::
     :: extensive simulation.                                         ::
     ::                                                               ::
     ::    The DNA test considers an alphabet of 4 letters::  C,G,A,T,::
     :: determined by two designated bits in the sequence of random   ::
     :: integers being tested.  It considers 10-letter words, so that ::
     :: as in OPSO and OQSO, there are 2^20 possible words, and the   ::
     :: mean number of missing words from a string of 2^21  (over-    ::
     :: lapping)  10-letter  words (2^21+9 "keystrokes") is 141909.   ::
```

```
    :: The standard deviation sigma=339 was determined as for OQSO   ::
    :: by simulation.  (Sigma for OPSO, 290, is the true value (to   ::
    :: three places), not determined by simulation.                  ::
    :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
OPSO test for generator block3.rng
 Output: No. missing words (mw), equiv normal variate (z), p-value (p)
                                                           mw     z     p
   OPSO for block3.rng      using bits 23 to 32        142294  1.326  .9077
   OPSO for block3.rng      using bits 22 to 31        141830  -.274  .3922
   OPSO for block3.rng      using bits 21 to 30        141767  -.491  .3118
   OPSO for block3.rng      using bits 20 to 29        141966   .195  .5775
   OPSO for block3.rng      using bits 19 to 28        142436  1.816  .9653
   OPSO for block3.rng      using bits 18 to 27        141697  -.732  .2320
   OPSO for block3.rng      using bits 17 to 26        141747  -.560  .2878
   OPSO for block3.rng      using bits 16 to 25        141758  -.522  .3009
   OPSO for block3.rng      using bits 15 to 24        141935   .089  .5353
   OPSO for block3.rng      using bits 14 to 23        141915   .020  .5078
   OPSO for block3.rng      using bits 13 to 22        141910   .002  .5009
   OPSO for block3.rng      using bits 12 to 21        141942   .113  .5449
   OPSO for block3.rng      using bits 11 to 20        141767  -.491  .3118
   OPSO for block3.rng      using bits 10 to 19        141807  -.353  .3621
   OPSO for block3.rng      using bits  9 to 18        141335 -1.980  .0238
   OPSO for block3.rng      using bits  8 to 17        141511 -1.374  .0848
   OPSO for block3.rng      using bits  7 to 16        141758  -.522  .3009
   OPSO for block3.rng      using bits  6 to 15        141903  -.022  .4913
   OPSO for block3.rng      using bits  5 to 14        142145   .813  .7918
   OPSO for block3.rng      using bits  4 to 13        142016   .368  .6435
   OPSO for block3.rng      using bits  3 to 12        142096   .644  .7401
   OPSO for block3.rng      using bits  2 to 11        142079   .585  .7208
   OPSO for block3.rng      using bits  1 to 10        141938   .099  .5394
OQSO test for generator block3.rng
 Output: No. missing words (mw), equiv normal variate (z), p-value (p)
                                                           mw     z     p
   OQSO for block3.rng      using bits 28 to 32        141980   .240  .5947
   OQSO for block3.rng      using bits 27 to 31        141943   .114  .5454
   OQSO for block3.rng      using bits 26 to 30        142140   .782  .7829
   OQSO for block3.rng      using bits 25 to 29        142045   .460  .6772
   OQSO for block3.rng      using bits 24 to 28        141602 -1.042  .1488
   OQSO for block3.rng      using bits 23 to 27        141965   .189  .5748
   OQSO for block3.rng      using bits 22 to 26        141830  -.269  .3940
   OQSO for block3.rng      using bits 21 to 25        141683  -.767  .2215
   OQSO for block3.rng      using bits 20 to 24        141580 -1.116  .1321
   OQSO for block3.rng      using bits 19 to 23        142123   .724  .7656
   OQSO for block3.rng      using bits 18 to 22        142102   .653  .7432
   OQSO for block3.rng      using bits 17 to 21        141844  -.221  .4124
   OQSO for block3.rng      using bits 16 to 20        142382  1.602  .9455
   OQSO for block3.rng      using bits 15 to 19        141566 -1.164  .1222
   OQSO for block3.rng      using bits 14 to 18        141869  -.137  .4456
   OQSO for block3.rng      using bits 13 to 17        141911   .006  .5023
   OQSO for block3.rng      using bits 12 to 16        142176   .904  .8170
   OQSO for block3.rng      using bits 11 to 15        142000   .307  .6207
   OQSO for block3.rng      using bits 10 to 14        141814  -.323  .3733
   OQSO for block3.rng      using bits  9 to 13        142282  1.263  .8968
   OQSO for block3.rng      using bits  8 to 12        142158   .843  .8004
   OQSO for block3.rng      using bits  7 to 11        142002   .314  .6233
   OQSO for block3.rng      using bits  6 to 10        141375 -1.811  .0350
   OQSO for block3.rng      using bits  5 to  9        141745  -.557  .2887
```

```
   OQSO for block3.rng      using bits  4 to  8       141719  -.645  .2594
   OQSO for block3.rng      using bits  3 to  7       141992   .280  .6104
   OQSO for block3.rng      using bits  2 to  6       141966   .192  .5762
   OQSO for block3.rng      using bits  1 to  5       142118   .707  .7603
  DNA test for generator block3.rng
  Output: No. missing words (mw), equiv normal variate (z), p-value (p)
                                                         mw     z     p
    DNA for block3.rng      using bits 31 to 32       141796  -.334  .3691
    DNA for block3.rng      using bits 30 to 31       141743  -.491  .3118
    DNA for block3.rng      using bits 29 to 30       141622  -.848  .1983
    DNA for block3.rng      using bits 28 to 29       141643  -.786  .2160
    DNA for block3.rng      using bits 27 to 28       141869  -.119  .4527
    DNA for block3.rng      using bits 26 to 27       142194   .840  .7995
    DNA for block3.rng      using bits 25 to 26       142230   .946  .8279
    DNA for block3.rng      using bits 24 to 25       141925   .046  .5184
    DNA for block3.rng      using bits 23 to 24       141860  -.146  .4422
    DNA for block3.rng      using bits 22 to 23       141837  -.213  .4155
    DNA for block3.rng      using bits 21 to 22       141860  -.146  .4422
    DNA for block3.rng      using bits 20 to 21       141683  -.668  .2522
    DNA for block3.rng      using bits 19 to 20       142454  1.607  .9459
    DNA for block3.rng      using bits 18 to 19       141938   .085  .5337
    DNA for block3.rng      using bits 17 to 18       141661  -.733  .2319
    DNA for block3.rng      using bits 16 to 17       141546 -1.072  .1419
    DNA for block3.rng      using bits 15 to 16       141922   .037  .5149
    DNA for block3.rng      using bits 14 to 15       141922   .037  .5149
    DNA for block3.rng      using bits 13 to 14       142113   .601  .7260
    DNA for block3.rng      using bits 12 to 13       141464 -1.314  .0945
    DNA for block3.rng      using bits 11 to 12       141567 -1.010  .1563
    DNA for block3.rng      using bits 10 to 11       141957   .141  .5559
    DNA for block3.rng      using bits  9 to 10       141672  -.700  .2419
    DNA for block3.rng      using bits  8 to  9       141588  -.948  .1716
    DNA for block3.rng      using bits  7 to  8       142070   .474  .6822
    DNA for block3.rng      using bits  6 to  7       142452  1.601  .9453
    DNA for block3.rng      using bits  5 to  6       142150   .710  .7611
    DNA for block3.rng      using bits  4 to  5       141867  -.125  .4503
    DNA for block3.rng      using bits  3 to  4       142282  1.099  .8642
    DNA for block3.rng      using bits  2 to  3       141848  -.181  .4282
    DNA for block3.rng      using bits  1 to  2       141186 -2.134  .0164

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

    :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
    ::     This is the COUNT-THE-1's TEST on a stream of bytes.      ::
    :: Consider the file under test as a stream of bytes (four per   ::
    :: 32 bit integer).  Each byte can contain from 0 to 8 1's,      ::
    :: with probabilities 1,8,28,56,70,56,28,8,1 over 256.  Now let  ::
    :: the stream of bytes provide a string of overlapping  5-letter ::
    :: words, each "letter" taking values A,B,C,D,E. The letters are ::
    :: determined by the number of 1's in a byte::  0,1,or 2 yield A,::
    :: 3 yields B, 4 yields C, 5 yields D and 6,7 or 8 yield E. Thus ::
    :: we have a monkey at a typewriter hitting five keys with vari- ::
    :: ous probabilities (37,56,70,56,37 over 256).  There are 5^5   ::
    :: possible 5-letter words, and from a string of 256,000 (over-  ::
    :: lapping) 5-letter words, counts are made on the frequencies   ::
    :: for each word.   The quadratic form in the weak inverse of    ::
    :: the covariance matrix of the cell counts provides a chisquare ::
    :: test::  Q5-Q4, the difference of the naive Pearson sums of    ::
```

```
     :: (OBS-EXP)^2/EXP on counts for 5- and 4-letter cell counts.    ::
     :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
   Test results for block3.rng
 Chi-square with 5^5-5^4=2500 d.of f. for sample size:2560000
                              chisquare  equiv normal  p-value
  Results fo COUNT-THE-1's in successive bytes:
 byte stream for block3.rng       2481.78      -.258      .398350
 byte stream for block3.rng       2530.09       .426      .664792

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

     :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
     ::     This is the COUNT-THE-1's TEST for specific bytes.        ::
     :: Consider the file under test as a stream of 32-bit integers.  ::
     :: From each integer, a specific byte is chosen , say the left-  ::
     :: most::  bits 1 to 8. Each byte can contain from 0 to 8 1's,   ::
     :: with probabilitie 1,8,28,56,70,56,28,8,1 over 256.  Now let   ::
     :: the specified bytes from successive integers provide a string ::
     :: of (overlapping) 5-letter words, each "letter" taking values  ::
     :: A,B,C,D,E. The letters are determined  by the number of 1's,  ::
     :: in that byte::  0,1,or 2 ---> A, 3 ---> B, 4 ---> C, 5 ---> D,::
     :: and  6,7 or 8 ---> E.  Thus we have a monkey at a typewriter  ::
     :: hitting five keys with with various probabilities::  37,56,70,::
     :: 56,37 over 256. There are 5^5 possible 5-letter words, and    ::
     :: from a string of 256,000 (overlapping) 5-letter words, counts ::
     :: are made on the frequencies for each word. The quadratic form ::
     :: in the weak inverse of the covariance matrix of the cell      ::
     :: counts provides a chisquare test::  Q5-Q4, the difference of  ::
     :: the naive Pearson  sums of (OBS-EXP)^2/EXP on counts for 5-   ::
     :: and 4-letter cell counts.                                     ::
     :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
 Chi-square with 5^5-5^4=2500 d.of f. for sample size: 256000
                     chisquare  equiv normal  p value
  Results for COUNT-THE-1's in specified bytes:
           bits  1 to  8  2502.17       .031      .512240
           bits  2 to  9  2391.41     -1.536      .062312
           bits  3 to 10  2478.35      -.306      .379749
           bits  4 to 11  2461.14      -.550      .291325
           bits  5 to 12  2466.26      -.477      .316638
           bits  6 to 13  2515.79       .223      .588350
           bits  7 to 14  2574.38      1.052      .853571
           bits  8 to 15  2595.48      1.350      .911533
           bits  9 to 16  2413.19     -1.228      .109779
           bits 10 to 17  2515.39       .218      .586149
           bits 11 to 18  2474.26      -.364      .357912
           bits 12 to 19  2446.21      -.761      .223429
           bits 13 to 20  2516.80       .238      .593913
           bits 14 to 21  2527.37       .387      .650665
           bits 15 to 22  2476.58      -.331      .370267
           bits 16 to 23  2406.41     -1.324      .092826
           bits 17 to 24  2445.97      -.764      .222393
           bits 18 to 25  2508.31       .118      .546792
           bits 19 to 26  2521.86       .309      .621399
           bits 20 to 27  2490.58      -.133      .446987
           bits 21 to 28  2561.37       .868      .807263
           bits 22 to 29  2470.35      -.419      .337491
           bits 23 to 30  2596.10      1.359      .912933
```

```
    bits 24 to 31  2476.07      -.338      .367532
    bits 25 to 32  2450.33      -.702      .241220

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
::               THIS IS A PARKING LOT TEST                       ::
:: In a square of side 100, randomly "park" a car---a circle of   ::
:: radius 1.   Then try to park a 2nd, a 3rd, and so on, each     ::
:: time parking "by ear".  That is, if an attempt to park a car   ::
:: causes a crash with one already parked, try again at a new     ::
:: random location. (To avoid path problems, consider parking     ::
:: helicopters rather than cars.)   Each attempt leads to either  ::
:: a crash or a success, the latter followed by an increment to   ::
:: the list of cars already parked. If we plot n:  the number of  ::
:: attempts, versus k::  the number successfully parked, we get a::
:: curve that should be similar to those provided by a perfect    ::
:: random number generator.  Theory for the behavior of such a    ::
:: random curve seems beyond reach, and as graphics displays are  ::
:: not available for this battery of tests, a simple characteriz  ::
:: ation of the random experiment is used: k, the number of cars  ::
:: successfully parked after n=12,000 attempts. Simulation shows  ::
:: that k should average 3523 with sigma 21.9 and is very close   ::
:: to normally distributed.  Thus (k-3523)/21.9 should be a st-   ::
:: andard normal variable, which, converted to a uniform varia-   ::
:: ble, provides input to a KSTEST based on a sample of 10.       ::
::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
      CDPARK: result of ten tests on file block3.rng
       Of 12,000 tries, the average no. of successes
            should be 3523 with sigma=21.9
       Successes: 3510    z-score:  -.594 p-value: .276387
       Successes: 3521    z-score:  -.091 p-value: .463618
       Successes: 3550    z-score:  1.233 p-value: .891189
       Successes: 3519    z-score:  -.183 p-value: .427537
       Successes: 3519    z-score:  -.183 p-value: .427537
       Successes: 3515    z-score:  -.365 p-value: .357445
       Successes: 3506    z-score:  -.776 p-value: .218799
       Successes: 3552    z-score:  1.324 p-value: .907282
       Successes: 3532    z-score:   .411 p-value: .659449
       Successes: 3514    z-score:  -.411 p-value: .340551

      square size   avg. no.  parked   sample sigma
        100.            3523.800       15.072
       KSTEST for the above 10: p=  .451665

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
::               THE MINIMUM DISTANCE TEST                        ::
:: It does this 100 times::   choose n=8000 random points in a    ::
:: square of side 10000.  Find d, the minimum distance between    ::
:: the (n^2-n)/2 pairs of points.  If the points are truly inde-  ::
:: pendent uniform, then d^2, the square of the minimum distance  ::
:: should be (very close to) exponentially distributed with mean  ::
:: .995 .  Thus 1-exp(-d^2/.995) should be uniform on [0,1) and   ::
:: a KSTEST on the resulting 100 values serves as a test of uni-  ::
:: formity for random points in the square. Test numbers=0 mod 5  ::
```

```
:: are printed but the KSTEST is based on the full set of 100    ::
:: random choices of 8000 points in the 10000x10000 square.      ::
:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
          This is the MINIMUM DISTANCE test
         for random integers in the file block3.rng
Sample no.    d^2     avg     equiv uni
     5     .3566   1.2487     .301211
    10     .2498   1.2662     .222060
    15    2.2682   1.1231     .897670
    20     .2259   1.1401     .203071
    25     .3029   1.1136     .262461
    30     .7571   1.0964     .532772
    35     .8437   1.0675     .571694
    40    1.1572   1.1584     .687463
    45     .2061   1.1185     .187072
    50    1.3004   1.0809     .729338
    55     .4219   1.0482     .345610
    60     .4818   1.0364     .383794
    65     .5123   1.0061     .402419
    70    1.1150    .9850     .673903
    75     .4943    .9704     .391531
    80    1.8742    .9571     .847955
    85     .1597    .9553     .148269
    90    2.0265    .9646     .869536
    95    1.3205    .9608     .734760
   100     .6081    .9716     .457292
MINIMUM DISTANCE TEST for block3.rng
     Result of KS test on 20 transformed mindist^2's:
                        p-value= .211625

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
::               THE 3DSPHERES TEST                              ::
:: Choose  4000 random points in a cube of edge 1000.  At each   ::
:: point, center a sphere large enough to reach the next closest ::
:: point. Then the volume of the smallest such sphere is (very   ::
:: close to) exponentially distributed with mean 120pi/3.  Thus  ::
:: the radius cubed is exponential with mean 30. (The mean is    ::
:: obtained by extensive simulation).  The 3DSPHERES test gener- ::
:: ates 4000 such spheres 20 times.  Each min radius cubed leads ::
:: to a uniform variable by means of 1-exp(-r^3/30.), then a     ::
::  KSTEST is done on the 20 p-values.                           ::
:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
          The 3DSPHERES test for file block3.rng
sample no:  1     r^3=  42.873     p-value= .76048
sample no:  2     r^3=  25.552     p-value= .57333
sample no:  3     r^3=  24.719     p-value= .56131
sample no:  4     r^3=   5.416     p-value= .16517
sample no:  5     r^3=  15.757     p-value= .40858
sample no:  6     r^3=  43.326     p-value= .76406
sample no:  7     r^3=   2.055     p-value= .06621
sample no:  8     r^3=  37.658     p-value= .71500
sample no:  9     r^3=  12.371     p-value= .33792
sample no: 10     r^3=  30.587     p-value= .63924
sample no: 11     r^3=  17.624     p-value= .44426
sample no: 12     r^3=  21.512     p-value= .51182
```

```
 sample no: 13     r^3=    .120     p-value= .00398
 sample no: 14     r^3=  18.439     p-value= .45917
 sample no: 15     r^3=  14.804     p-value= .38949
 sample no: 16     r^3=  32.237     p-value= .65856
 sample no: 17     r^3=  86.750     p-value= .94452
 sample no: 18     r^3=   6.576     p-value= .19683
 sample no: 19     r^3=   5.668     p-value= .17216
 sample no: 20     r^3=   8.877     p-value= .25614
  A KS test is applied to those 20 p-values.
---------------------------------------------------------
       3DSPHERES test for file block3.rng          p-value= .411318
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
     ::      This is the SQEEZE test                                 ::
     ::  Random integers are floated to get uniforms on [0,1). Start- ::
     ::  ing with k=2^31=2147483647, the test finds j, the number of  ::
     ::  iterations necessary to reduce k to 1, using the reduction   ::
     ::  k=ceiling(k*U), with U provided by floating integers from    ::
     ::  the file being tested.  Such j's are found 100,000 times,    ::
     ::  then counts for the number of times j was <=6,7,...,47,>=48  ::
     ::  are used to provide a chi-square test for cell frequencies.  ::
     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
           RESULTS OF SQUEEZE TEST FOR block3.rng
         Table of standardized frequency counts
     ( (obs-exp)/sqrt(exp) )^2
        for j taking values <=6,7,8,...,47,>=48:
      .6       .1      -.8      -.7       .2      1.2
     -.6      -.8       .9     -1.3      -.6      -.1
     -.2      -.6      1.5       .0     -1.0       .0
      .6     -1.1       .3       .2     -1.0       .0
      .6      -.5      1.1       .7      2.0       .8
      .7      -.2       .1      2.1       .8      -.5
     -.7      -.1     -1.6     -1.3       .9      2.0
      .8
           Chi-square with 42 degrees of freedom: 36.324
              z-score=  -.619  p-value= .282057
______________________________________________________________

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
     ::             The  OVERLAPPING SUMS test                       ::
     :: Integers are floated to get a sequence U(1),U(2),... of uni-  ::
     :: form [0,1) variables.  Then overlapping sums,                 ::
     ::   S(1)=U(1)+...+U(100), S2=U(2)+...+U(101),... are formed.    ::
     :: The S's are virtually normal with a certain covariance mat-   ::
     :: rix.  A linear transformation of the S's converts them to a   ::
     :: sequence of independent standard normals, which are converted ::
     :: to uniform variables for a KSTEST. The  p-values from ten     ::
     :: KSTESTs are given still another KSTEST.                       ::
     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
                Test no.  1      p-value  .864680
                Test no.  2      p-value  .519832
                Test no.  3      p-value  .883525
                Test no.  4      p-value  .860066
                Test no.  5      p-value  .519336
```

```
              Test no.  6      p-value  .526971
              Test no.  7      p-value  .842970
              Test no.  8      p-value  .611857
              Test no.  9      p-value  .540477
              Test no. 10      p-value  .350981
   Results of the OSUM test for block3.rng
        KSTEST on the above 10 p-values:  .913041

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
     ::     This is the RUNS test.  It counts runs up, and runs down, ::
     :: in a sequence of uniform [0,1) variables, obtained by float-  ::
     :: ing the 32-bit integers in the specified file. This example   ::
     :: shows how runs are counted:  .123,.357,.789,.425,.224,.416,.95::
     :: contains an up-run of length 3, a down-run of length 2 and an ::
     :: up-run of (at least) 2, depending on the next values.  The    ::
     :: covariance matrices for the runs-up and runs-down are well    ::
     :: known, leading to chisquare tests for quadratic forms in the  ::
     :: weak inverses of the covariance matrices.  Runs are counted   ::
     :: for sequences of length 10,000.  This is done ten times. Then ::
     :: repeated.                                                     ::
     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
           The RUNS test for file block3.rng
     Up and down runs in a sample of 10000
_________________________________________________
                Run test for block3.rng     :
       runs up; ks test for 10 p's: .414940
     runs down; ks test for 10 p's: .276074
                Run test for block3.rng     :
       runs up; ks test for 10 p's: .695511
     runs down; ks test for 10 p's: .206234

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
     :: This is the CRAPS TEST. It plays 200,000 games of craps, finds::
     :: the number of wins and the number of throws necessary to end  ::
     :: each game.  The number of wins should be (very close to) a    ::
     :: normal with mean 200000p and variance 200000p(1-p), with      ::
     :: p=244/495.  Throws necessary to complete the game can vary    ::
     :: from 1 to infinity, but counts for all>21 are lumped with 21. ::
     :: A chi-square test is made on the no.-of-throws cell counts.   ::
     :: Each 32-bit integer from the test file provides the value for ::
     :: the throw of a die, by floating to [0,1), multiplying by 6    ::
     :: and taking 1 plus the integer part of the result.             ::
     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
              Results of craps test for block3.rng
  No. of wins:  Observed Expected
                                98657    98585.86
                98657= No. of wins, z-score=  .318 pvalue= .62483
   Analysis of Throws-per-Game:
 Chisq=  27.83 for 20 degrees of freedom, p=  .88641
             Throws Observed Expected  Chisq     Sum
                1    66161    66666.7   3.836    3.836
                2    37705    37654.3    .068    3.904
                3    27050    26954.7    .337    4.240
```

```
                   4    19665    19313.5   6.399   10.639
                   5    13731    13851.4   1.047   11.686
                   6     9942     9943.5    .000   11.686
                   7     7167     7145.0    .068   11.754
                   8     5138     5139.1    .000   11.754
                   9     3800     3699.9   2.710   14.464
                  10     2669     2666.3    .003   14.467
                  11     1976     1923.3   1.442   15.909
                  12     1391     1388.7    .004   15.913
                  13      971     1003.7   1.066   16.979
                  14      666      726.1   4.981   21.960
                  15      532      525.8    .072   22.033
                  16      411      381.2   2.338   24.370
                  17      268      276.5    .264   24.634
                  18      203      200.8    .023   24.657
                  19      166      146.0   2.744   27.402
                  20      110      106.2    .135   27.536
                  21      278      287.1    .289   27.826
            SUMMARY  FOR block3.rng
                p-value for no. of wins: .624827
                p-value for throws/game: .886415

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

 Results of DIEHARD battery of tests sent to file report3.txt



       NOTE: Most of the tests in DIEHARD return a p-value, which
       should be uniform on [0,1) if the input file contains truly
       independent random bits.   Those p-values are obtained by
       p=F(X), where F is the assumed distribution of the sample
       random variable X---often normal. But that assumed F is just
       an asymptotic approximation, for which the fit will be worst
       in the tails. Thus you should not be surprised with
       occasional p-values near 0 or 1, such as .0012 or .9983.
       When a bit stream really FAILS BIG, you will get p's of 0 or
       1 to six or more places.  By all means, do not, as a
       Statistician might, think that a p < .025 or p> .975 means
       that the RNG has "failed the test at the .05 level".  Such
       p's happen among the hundreds that DIEHARD produces, even
       with good RNG's.  So keep in mind that " p happens".
     :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
     ::            This is the BIRTHDAY SPACINGS TEST                 ::
     :: Choose m birthdays in a year of n days.  List the spacings    ::
     :: between the birthdays.  If j is the number of values that     ::
     :: occur more than once in that list, then j is asymptotically   ::
     :: Poisson distributed with mean m^3/(4n).  Experience shows n   ::
     :: must be quite large, say n>=2^18, for comparing the results   ::
     :: to the Poisson distribution with that mean.  This test uses   ::
     :: n=2^24 and m=2^9,  so that the underlying distribution for j  ::
     :: is taken to be Poisson with lambda=2^27/(2^26)=2.  A sample   ::
     :: of 500 j's is taken, and a chi-square goodness of fit test    ::
     :: provides a p value.  The first test uses bits 1-24 (counting  ::
     :: from the left) from integers in the specified file.           ::
     ::   Then the file is closed and reopened. Next, bits 2-25 are   ::
```

```
    :: used to provide birthdays, then 3-26 and so on to bits 9-32.  ::
    :: Each set of bits provides a p-value, and the nine p-values    ::
    :: provide a sample for a KSTEST.                                ::
    ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
BIRTHDAY SPACINGS TEST, M= 512 N=2**24 LAMBDA=  2.0000
          Results for block4.rng
                  For a sample of size 500:     mean
          block4.rng      using bits  1 to 24   2.042
 duplicate       number       number
 spacings       observed     expected
       0          69.       67.668
       1         147.      135.335
       2         104.      135.335
       3         107.       90.224
       4          38.       45.112
       5          23.       18.045
 6 to INF         12.        8.282
Chisquare with  6 d.o.f. =    15.56 p-value=  .983664
 :::::::::::::::::::::::::::::::::::::::::
                  For a sample of size 500:     mean
          block4.rng      using bits  2 to 25   1.932
 duplicate       number       number
 spacings       observed     expected
       0          75.       67.668
       1         136.      135.335
       2         134.      135.335
       3          95.       90.224
       4          35.       45.112
       5          16.       18.045
 6 to INF          9.        8.282
Chisquare with  6 d.o.f. =     3.62 p-value=  .272641
 :::::::::::::::::::::::::::::::::::::::::
                  For a sample of size 500:     mean
          block4.rng      using bits  3 to 26   1.920
 duplicate       number       number
 spacings       observed     expected
       0          76.       67.668
       1         140.      135.335
       2         135.      135.335
       3          78.       90.224
       4          52.       45.112
       5           9.       18.045
 6 to INF         10.        8.282
Chisquare with  6 d.o.f. =     8.79 p-value=  .813994
 :::::::::::::::::::::::::::::::::::::::::
                  For a sample of size 500:     mean
          block4.rng      using bits  4 to 27   2.114
 duplicate       number       number
 spacings       observed     expected
       0          58.       67.668
       1         122.      135.335
       2         143.      135.335
       3         102.       90.224
       4          50.       45.112
       5          15.       18.045
 6 to INF         10.        8.282
Chisquare with  6 d.o.f. =     6.07 p-value=  .584197
```

```
 :::::::::::::::::::::::::::::::::::::::::
                 For a sample of size 500:     mean
         block4.rng      using bits  5 to 28   1.988
 duplicate       number       number
 spacings       observed     expected
      0          58.       67.668
      1         150.      135.335
      2         136.      135.335
      3          85.       90.224
      4          49.       45.112
      5          13.       18.045
 6 to INF         9.        8.282
Chisquare with  6 d.o.f. =     5.08 p-value=  .466874
 :::::::::::::::::::::::::::::::::::::::::
                 For a sample of size 500:     mean
         block4.rng      using bits  6 to 29   1.976
 duplicate       number       number
 spacings       observed     expected
      0          57.       67.668
      1         148.      135.335
      2         132.      135.335
      3         105.       90.224
      4          37.       45.112
      5          15.       18.045
 6 to INF         6.        8.282
Chisquare with  6 d.o.f. =     7.97 p-value=  .759703
 :::::::::::::::::::::::::::::::::::::::::
                 For a sample of size 500:     mean
         block4.rng      using bits  7 to 30   1.950
 duplicate       number       number
 spacings       observed     expected
      0          72.       67.668
      1         130.      135.335
      2         140.      135.335
      3         100.       90.224
      4          35.       45.112
      5          14.       18.045
 6 to INF         9.        8.282
Chisquare with  6 d.o.f. =     4.94 p-value=  .448894
 :::::::::::::::::::::::::::::::::::::::::
                 For a sample of size 500:     mean
         block4.rng      using bits  8 to 31   2.024
 duplicate       number       number
 spacings       observed     expected
      0          65.       67.668
      1         141.      135.335
      2         126.      135.335
      3          88.       90.224
      4          55.       45.112
      5          18.       18.045
 6 to INF         7.        8.282
Chisquare with  6 d.o.f. =     3.41 p-value=  .243695
 :::::::::::::::::::::::::::::::::::::::::
                 For a sample of size 500:     mean
         block4.rng      using bits  9 to 32   2.116
 duplicate       number       number
 spacings       observed     expected
```

```
        0          56.       67.668
        1         134.      135.335
        2         141.      135.335
        3          86.       90.224
        4          48.       45.112
        5          24.       18.045
  6 to INF         11.        8.282
 Chisquare with  6 d.o.f. =     5.50 p-value=  .518819
  :::::::::::::::::::::::::::::::::::::::::
   The 9 p-values were
        .983664   .272641   .813994   .584197   .466874
        .759703   .448894   .243695   .518819
  A KSTEST for the 9 p-values yields  .373832

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

     :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
     ::            THE OVERLAPPING 5-PERMUTATION TEST                 ::
     :: This is the OPERM5 test.  It looks at a sequence of one mill- ::
     :: ion 32-bit random integers.  Each set of five consecutive     ::
     :: integers can be in one of 120 states, for the 5! possible or- ::
     :: derings of five numbers.  Thus the 5th, 6th, 7th,...numbers   ::
     :: each provide a state. As many thousands of state transitions  ::
     :: are observed,  cumulative counts are made of the number of    ::
     :: occurences of each state.  Then the quadratic form in the     ::
     :: weak inverse of the 120x120 covariance matrix yields a test   ::
     :: equivalent to the likelihood ratio test that the 120 cell     ::
     :: counts came from the specified (asymptotically) normal dis-   ::
     :: tribution with the specified 120x120 covariance matrix (with  ::
     :: rank 99).  This version uses 1,000,000 integers, twice.       ::
     :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
           OPERM5 test for file block4.rng
     For a sample of 1,000,000 consecutive 5-tuples,
 chisquare for 99 degrees of freedom=106.939; p-value= .724796
           OPERM5 test for file block4.rng
     For a sample of 1,000,000 consecutive 5-tuples,
 chisquare for 99 degrees of freedom= 98.789; p-value= .512905
     :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
     :: This is the BINARY RANK TEST for 31x31 matrices. The leftmost ::
     :: 31 bits of 31 random integers from the test sequence are used ::
     :: to form a 31x31 binary matrix over the field {0,1}. The rank  ::
     :: is determined. That rank can be from 0 to 31, but ranks< 28   ::
     :: are rare, and their counts are pooled with those for rank 28. ::
     :: Ranks are found for 40,000 such random matrices and a chisqua-::
     :: re test is performed on counts for ranks 31,30,29 and <=28.   ::
     :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
    Binary rank test for block4.rng
         Rank test for 31x31 binary matrices:
        rows from leftmost 31 bits of each 32-bit integer
      rank   observed  expected (o-e)^2/e  sum
        28       216     211.4   .099304     .099
        29      5054    5134.0  1.246908    1.346
        30     23077   23103.0   .029366    1.376
        31     11653   11551.5   .891423    2.267
  chisquare= 2.267 for 3 d. of f.; p-value= .547206
--------------------------------------------------------------
     :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
```

```
     :: This is the BINARY RANK TEST for 32x32 matrices. A random 32x ::
     :: 32 binary matrix is formed, each row a 32-bit random integer. ::
     :: The rank is determined. That rank can be from 0 to 32, ranks  ::
     :: less than 29 are rare, and their counts are pooled with those ::
     :: for rank 29.  Ranks are found for 40,000 such random matrices ::
     :: and a chisquare test is performed on counts for ranks  32,31, ::
     :: 30 and <=29.                                                  ::
     :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
    Binary rank test for block4.rng
         Rank test for 32x32 binary matrices:
        rows from leftmost 32 bits of each 32-bit integer
      rank   observed  expected (o-e)^2/e  sum
        29       220     211.4   .348364     .348
        30      5067    5134.0   .874633    1.223
        31     23016   23103.0   .327972    1.551
        32     11697   11551.5  1.832065    3.383
  chisquare= 3.383 for 3 d. of f.; p-value= .696684
--------------------------------------------------------------

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

     :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
     :: This is the BINARY RANK TEST for 6x8 matrices.  From each of  ::
     :: six random 32-bit integers from the generator under test, a   ::
     :: specified byte is chosen, and the resulting six bytes form a  ::
     :: 6x8 binary matrix whose rank is determined.  That rank can be ::
     :: from 0 to 6, but ranks 0,1,2,3 are rare; their counts are     ::
     :: pooled with those for rank 4. Ranks are found for 100,000     ::
     :: random matrices, and a chi-square test is performed on        ::
     :: counts for ranks 6,5 and <=4.                                 ::
     :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
         Binary Rank Test for block4.rng
        Rank of a 6x8 binary matrix,
     rows formed from eight bits of the RNG block4.rng
     b-rank test for bits  1 to  8
                     OBSERVED   EXPECTED     (O-E)^2/E      SUM
          r<=4          997       944.3       2.941       2.941
          r =5        21793     21743.9        .111       3.052
          r =6        77210     77311.8        .134       3.186
                        p=1-exp(-SUM/2)= .79667
        Rank of a 6x8 binary matrix,
     rows formed from eight bits of the RNG block4.rng
     b-rank test for bits  2 to  9
                     OBSERVED   EXPECTED     (O-E)^2/E      SUM
          r<=4          952       944.3        .063        .063
          r =5        21798     21743.9        .135        .197
          r =6        77250     77311.8        .049        .247
                        p=1-exp(-SUM/2)= .11608
        Rank of a 6x8 binary matrix,
     rows formed from eight bits of the RNG block4.rng
     b-rank test for bits  3 to 10
                     OBSERVED   EXPECTED     (O-E)^2/E      SUM
          r<=4          929       944.3        .248        .248
          r =5        22135     21743.9       7.035       7.283
          r =6        76936     77311.8       1.827       9.109
                        p=1-exp(-SUM/2)= .98948
        Rank of a 6x8 binary matrix,
```

```
rows formed from eight bits of the RNG block4.rng
b-rank test for bits  4 to 11
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          985       944.3       1.754        1.754
     r =5        22045     21743.9       4.169        5.924
     r =6        76970     77311.8       1.511        7.435
                   p=1-exp(-SUM/2)= .97570
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block4.rng
b-rank test for bits  5 to 12
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          979       944.3       1.275        1.275
     r =5        21831     21743.9        .349        1.624
     r =6        77190     77311.8        .192        1.816
                   p=1-exp(-SUM/2)= .59663
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block4.rng
b-rank test for bits  6 to 13
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          959       944.3        .229         .229
     r =5        21977     21743.9       2.499        2.728
     r =6        77064     77311.8        .794        3.522
                   p=1-exp(-SUM/2)= .82812
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block4.rng
b-rank test for bits  7 to 14
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          999       944.3       3.168        3.168
     r =5        21794     21743.9        .115        3.284
     r =6        77207     77311.8        .142        3.426
                   p=1-exp(-SUM/2)= .81967
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block4.rng
b-rank test for bits  8 to 15
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          902       944.3       1.895        1.895
     r =5        21537     21743.9       1.969        3.864
     r =6        77561     77311.8        .803        4.667
                   p=1-exp(-SUM/2)= .90304
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block4.rng
b-rank test for bits  9 to 16
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          898       944.3       2.270        2.270
     r =5        21536     21743.9       1.988        4.258
     r =6        77566     77311.8        .836        5.094
                   p=1-exp(-SUM/2)= .92168
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block4.rng
b-rank test for bits 10 to 17
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          979       944.3       1.275        1.275
     r =5        21540     21743.9       1.912        3.187
     r =6        77481     77311.8        .370        3.557
                   p=1-exp(-SUM/2)= .83114
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block4.rng
```

b-rank test for bits 11 to 18

	OBSERVED	EXPECTED	(O-E)^2/E	SUM
r<=4	981	944.3	1.426	1.426
r =5	21722	21743.9	.022	1.448
r =6	77297	77311.8	.003	1.451

p=1-exp(-SUM/2)= .51595

Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block4.rng
b-rank test for bits 12 to 19

	OBSERVED	EXPECTED	(O-E)^2/E	SUM
r<=4	920	944.3	.625	.625
r =5	21889	21743.9	.968	1.594
r =6	77191	77311.8	.189	1.782

p=1-exp(-SUM/2)= .58984

Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block4.rng
b-rank test for bits 13 to 20

	OBSERVED	EXPECTED	(O-E)^2/E	SUM
r<=4	891	944.3	3.009	3.009
r =5	21939	21743.9	1.751	4.759
r =6	77170	77311.8	.260	5.019

p=1-exp(-SUM/2)= .91870

Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block4.rng
b-rank test for bits 14 to 21

	OBSERVED	EXPECTED	(O-E)^2/E	SUM
r<=4	887	944.3	3.477	3.477
r =5	21797	21743.9	.130	3.607
r =6	77316	77311.8	.000	3.607

p=1-exp(-SUM/2)= .83528

Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block4.rng
b-rank test for bits 15 to 22

	OBSERVED	EXPECTED	(O-E)^2/E	SUM
r<=4	986	944.3	1.841	1.841
r =5	21688	21743.9	.144	1.985
r =6	77326	77311.8	.003	1.988

p=1-exp(-SUM/2)= .62984

Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block4.rng
b-rank test for bits 16 to 23

	OBSERVED	EXPECTED	(O-E)^2/E	SUM
r<=4	989	944.3	2.116	2.116
r =5	22011	21743.9	3.281	5.397
r =6	77000	77311.8	1.258	6.654

p=1-exp(-SUM/2)= .96411

Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block4.rng
b-rank test for bits 17 to 24

	OBSERVED	EXPECTED	(O-E)^2/E	SUM
r<=4	995	944.3	2.722	2.722
r =5	21909	21743.9	1.254	3.976
r =6	77096	77311.8	.602	4.578

p=1-exp(-SUM/2)= .89863

Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block4.rng
b-rank test for bits 18 to 25

	OBSERVED	EXPECTED	(O-E)^2/E	SUM
r<=4	947	944.3	.008	.008
r =5	21764	21743.9	.019	.026
r =6	77289	77311.8	.007	.033

p=1-exp(-SUM/2)= .01637

Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block4.rng
b-rank test for bits 19 to 26

	OBSERVED	EXPECTED	(O-E)^2/E	SUM
r<=4	966	944.3	.499	.499
r =5	21799	21743.9	.140	.638
r =6	77235	77311.8	.076	.715

p=1-exp(-SUM/2)= .30041

Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block4.rng
b-rank test for bits 20 to 27

	OBSERVED	EXPECTED	(O-E)^2/E	SUM
r<=4	893	944.3	2.787	2.787
r =5	21930	21743.9	1.593	4.380
r =6	77177	77311.8	.235	4.615

p=1-exp(-SUM/2)= .90048

Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block4.rng
b-rank test for bits 21 to 28

	OBSERVED	EXPECTED	(O-E)^2/E	SUM
r<=4	924	944.3	.436	.436
r =5	21827	21743.9	.318	.754
r =6	77249	77311.8	.051	.805

p=1-exp(-SUM/2)= .33137

Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block4.rng
b-rank test for bits 22 to 29

	OBSERVED	EXPECTED	(O-E)^2/E	SUM
r<=4	953	944.3	.080	.080
r =5	21853	21743.9	.547	.628
r =6	77194	77311.8	.180	.807

p=1-exp(-SUM/2)= .33204

Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block4.rng
b-rank test for bits 23 to 30

	OBSERVED	EXPECTED	(O-E)^2/E	SUM
r<=4	974	944.3	.934	.934
r =5	21770	21743.9	.031	.965
r =6	77256	77311.8	.040	1.006

p=1-exp(-SUM/2)= .39518

Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block4.rng
b-rank test for bits 24 to 31

	OBSERVED	EXPECTED	(O-E)^2/E	SUM
r<=4	924	944.3	.436	.436
r =5	21633	21743.9	.566	1.002
r =6	77443	77311.8	.223	1.225

p=1-exp(-SUM/2)= .45793

Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block4.rng
b-rank test for bits 25 to 32

	OBSERVED	EXPECTED	(O-E)^2/E	SUM

```
         r<=4          974       944.3        .934        .934
         r =5        21564     21743.9       1.488       2.422
         r =6        77462     77311.8        .292       2.714
                        p=1-exp(-SUM/2)= .74260
   TEST SUMMARY, 25 tests on 100,000 random 6x8 matrices
 These should be 25 uniform [0,1] random variables:
     .796674     .116079     .989482     .975702     .596633
     .828123     .819670     .903039     .921677     .831138
     .515949     .589840     .918701     .835279     .629844
     .964106     .898630     .016374     .300413     .900484
     .331373     .332036     .395180     .457927     .742600
   brank test summary for block4.rng
       The KS test for those 25 supposed UNI's yields
                    KS p-value= .995650

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
     ::                   THE BITSTREAM TEST                         ::
     :: The file under test is viewed as a stream of bits. Call them  ::
     :: b1,b2,... .  Consider an alphabet with two "letters", 0 and 1 ::
     :: and think of the stream of bits as a succession of 20-letter  ::
     :: "words", overlapping.  Thus the first word is b1b2...b20, the ::
     :: second is b2b3...b21, and so on.  The bitstream test counts   ::
     :: the number of missing 20-letter (20-bit) words in a string of ::
     :: 2^21 overlapping 20-letter words.  There are 2^20 possible 20 ::
     :: letter words.  For a truly random string of 2^21+19 bits, the ::
     :: number of missing words j should be (very close to) normally  ::
     :: distributed with mean 141,909 and sigma 428.  Thus            ::
     ::  (j-141909)/428 should be a standard normal variate (z score) ::
     :: that leads to a uniform [0,1) p value.  The test is repeated  ::
     :: twenty times.                                                 ::
     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
 THE OVERLAPPING 20-tuples BITSTREAM  TEST, 20 BITS PER WORD, N words
   This test uses N=2^21 and samples the bitstream 20 times.
  No. missing words should average  141909. with sigma=428.
---------------------------------------------------------
 tst no  1:  141285 missing words,   -1.46 sigmas from mean, p-value= .07232
 tst no  2:  141829 missing words,    -.19 sigmas from mean, p-value= .42556
 tst no  3:  141483 missing words,   -1.00 sigmas from mean, p-value= .15960
 tst no  4:  141640 missing words,    -.63 sigmas from mean, p-value= .26459
 tst no  5:  141890 missing words,    -.05 sigmas from mean, p-value= .48199
 tst no  6:  142712 missing words,    1.88 sigmas from mean, p-value= .96963
 tst no  7:  142068 missing words,     .37 sigmas from mean, p-value= .64458
 tst no  8:  141390 missing words,   -1.21 sigmas from mean, p-value= .11249
 tst no  9:  141436 missing words,   -1.11 sigmas from mean, p-value= .13438
 tst no 10:  141522 missing words,    -.90 sigmas from mean, p-value= .18274
 tst no 11:  142186 missing words,     .65 sigmas from mean, p-value= .74100
 tst no 12:  141419 missing words,   -1.15 sigmas from mean, p-value= .12597
 tst no 13:  141850 missing words,    -.14 sigmas from mean, p-value= .44488
 tst no 14:  142004 missing words,     .22 sigmas from mean, p-value= .58753
 tst no 15:  141425 missing words,   -1.13 sigmas from mean, p-value= .12890
 tst no 16:  142197 missing words,     .67 sigmas from mean, p-value= .74925
 tst no 17:  141595 missing words,    -.73 sigmas from mean, p-value= .23135
 tst no 18:  141883 missing words,    -.06 sigmas from mean, p-value= .47547
 tst no 19:  141233 missing words,   -1.58 sigmas from mean, p-value= .05703
 tst no 20:  141844 missing words,    -.15 sigmas from mean, p-value= .43934
```

```
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
     ::             The tests OPSO, OQSO and DNA                     ::
     ::         OPSO means Overlapping-Pairs-Sparse-Occupancy        ::
     :: The OPSO test considers 2-letter words from an alphabet of   ::
     :: 1024 letters.  Each letter is determined by a specified ten  ::
     :: bits from a 32-bit integer in the sequence to be tested. OPSO ::
     :: generates  2^21 (overlapping) 2-letter words  (from 2^21+1   ::
     :: "keystrokes")  and counts the number of missing words---that ::
     :: is 2-letter words which do not appear in the entire sequence. ::
     :: That count should be very close to normally distributed with ::
     :: mean 141,909, sigma 290. Thus (missingwrds-141909)/290 should ::
     :: be a standard normal variable. The OPSO test takes 32 bits at ::
     :: a time from the test file and uses a designated set of ten   ::
     :: consecutive bits. It then restarts the file for the next de- ::
     :: signated 10 bits, and so on.                                 ::
     ::                                                              ::
     ::     OQSO means Overlapping-Quadruples-Sparse-Occupancy       ::
     ::   The test OQSO is similar, except that it considers 4-letter ::
     :: words from an alphabet of 32 letters, each letter determined ::
     :: by a designated string of 5 consecutive bits from the test   ::
     :: file, elements of which are assumed 32-bit random integers.  ::
     :: The mean number of missing words in a sequence of 2^21 four- ::
     :: letter words,  (2^21+3 "keystrokes"), is again 141909, with  ::
     :: sigma = 295.  The mean is based on theory; sigma comes from  ::
     :: extensive simulation.                                        ::
     ::                                                              ::
     ::    The DNA test considers an alphabet of 4 letters::  C,G,A,T,::
     :: determined by two designated bits in the sequence of random  ::
     :: integers being tested.  It considers 10-letter words, so that ::
     :: as in OPSO and OQSO, there are 2^20 possible words, and the  ::
     :: mean number of missing words from a string of 2^21  (over-   ::
     :: lapping)  10-letter  words (2^21+9 "keystrokes") is 141909.  ::
     :: The standard deviation sigma=339 was determined as for OQSO  ::
     :: by simulation.  (Sigma for OPSO, 290, is the true value (to  ::
     :: three places), not determined by simulation.                 ::
     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
 OPSO test for generator block4.rng
  Output: No. missing words (mw), equiv normal variate (z), p-value (p)
                                                           mw     z     p
    OPSO for block4.rng      using bits 23 to 32        142673  2.633  .9958
    OPSO for block4.rng      using bits 22 to 31        142082   .595  .7242
    OPSO for block4.rng      using bits 21 to 30        141722  -.646  .2592
    OPSO for block4.rng      using bits 20 to 29        141657  -.870  .1921
    OPSO for block4.rng      using bits 19 to 28        142015   .364  .6422
    OPSO for block4.rng      using bits 18 to 27        141571 -1.167  .1217
    OPSO for block4.rng      using bits 17 to 26        141972   .216  .5855
    OPSO for block4.rng      using bits 16 to 25        142409  1.723  .9576
    OPSO for block4.rng      using bits 15 to 24        142098   .651  .7423
    OPSO for block4.rng      using bits 14 to 23        142127   .751  .7736
    OPSO for block4.rng      using bits 13 to 22        142402  1.699  .9553
    OPSO for block4.rng      using bits 12 to 21        141893  -.056  .4775
    OPSO for block4.rng      using bits 11 to 20        142448  1.857  .9684
    OPSO for block4.rng      using bits 10 to 19        142081   .592  .7231
    OPSO for block4.rng      using bits  9 to 18        141887  -.077  .4693
```

```
   OPSO for block4.rng      using bits  8 to 17       141902  -.025  .4899
   OPSO for block4.rng      using bits  7 to 16       141934   .085  .5339
   OPSO for block4.rng      using bits  6 to 15       141341 -1.960  .0250
   OPSO for block4.rng      using bits  5 to 14       141717  -.663  .2536
   OPSO for block4.rng      using bits  4 to 13       142229  1.102  .8648
   OPSO for block4.rng      using bits  3 to 12       141961   .178  .5707
   OPSO for block4.rng      using bits  2 to 11       142307  1.371  .9149
   OPSO for block4.rng      using bits  1 to 10       141882  -.094  .4625
OQSO test for generator block4.rng
 Output: No. missing words (mw), equiv normal variate (z), p-value (p)
                                                         mw     z     p
   OQSO for block4.rng      using bits 28 to 32       141386 -1.774  .0380
   OQSO for block4.rng      using bits 27 to 31       141746  -.554  .2899
   OQSO for block4.rng      using bits 26 to 30       142032   .416  .6612
   OQSO for block4.rng      using bits 25 to 29       141764  -.493  .3111
   OQSO for block4.rng      using bits 24 to 28       141991   .277  .6091
   OQSO for block4.rng      using bits 23 to 27       141616  -.994  .1600
   OQSO for block4.rng      using bits 22 to 26       141331 -1.960  .0250
   OQSO for block4.rng      using bits 21 to 25       141591 -1.079  .1403
   OQSO for block4.rng      using bits 20 to 24       141644  -.899  .1842
   OQSO for block4.rng      using bits 19 to 23       142023   .385  .6500
   OQSO for block4.rng      using bits 18 to 22       141337 -1.940  .0262
   OQSO for block4.rng      using bits 17 to 21       142304  1.338  .9095
   OQSO for block4.rng      using bits 16 to 20       142022   .382  .6487
   OQSO for block4.rng      using bits 15 to 19       141933   .080  .5320
   OQSO for block4.rng      using bits 14 to 18       141810  -.337  .3682
   OQSO for block4.rng      using bits 13 to 17       142403  1.673  .9529
   OQSO for block4.rng      using bits 12 to 16       142252  1.162  .8773
   OQSO for block4.rng      using bits 11 to 15       142016   .362  .6412
   OQSO for block4.rng      using bits 10 to 14       141487 -1.432  .0761
   OQSO for block4.rng      using bits  9 to 13       141672  -.805  .2106
   OQSO for block4.rng      using bits  8 to 12       141431 -1.621  .0525
   OQSO for block4.rng      using bits  7 to 11       141849  -.205  .4190
   OQSO for block4.rng      using bits  6 to 10       141523 -1.310  .0952
   OQSO for block4.rng      using bits  5 to  9       142546  2.158  .9845
   OQSO for block4.rng      using bits  4 to  8       142079   .575  .7174
   OQSO for block4.rng      using bits  3 to  7       141920   .036  .5144
   OQSO for block4.rng      using bits  2 to  6       142015   .358  .6399
   OQSO for block4.rng      using bits  1 to  5       141813  -.327  .3720
 DNA test for generator block4.rng
 Output: No. missing words (mw), equiv normal variate (z), p-value (p)
                                                         mw     z     p
    DNA for block4.rng      using bits 31 to 32       142118   .616  .7309
    DNA for block4.rng      using bits 30 to 31       142456  1.613  .9466
    DNA for block4.rng      using bits 29 to 30       141915   .017  .5067
    DNA for block4.rng      using bits 28 to 29       141888  -.063  .4749
    DNA for block4.rng      using bits 27 to 28       142278  1.088  .8616
    DNA for block4.rng      using bits 26 to 27       142398  1.442  .9253
    DNA for block4.rng      using bits 25 to 26       141698  -.623  .2665
    DNA for block4.rng      using bits 24 to 25       142306  1.170  .8790
    DNA for block4.rng      using bits 23 to 24       142123   .630  .7358
    DNA for block4.rng      using bits 22 to 23       142329  1.238  .8921
    DNA for block4.rng      using bits 21 to 22       141377 -1.570  .0582
    DNA for block4.rng      using bits 20 to 21       142162   .745  .7720
    DNA for block4.rng      using bits 19 to 20       141743  -.491  .3118
    DNA for block4.rng      using bits 18 to 19       141916   .020  .5079
    DNA for block4.rng      using bits 17 to 18       142169   .766  .7782
```

```
     DNA for block4.rng      using bits 16 to 17        141629  -.827  .2041
     DNA for block4.rng      using bits 15 to 16        141782  -.376  .3536
     DNA for block4.rng      using bits 14 to 15        142153   .719  .7639
     DNA for block4.rng      using bits 13 to 14        142201   .860  .8052
     DNA for block4.rng      using bits 12 to 13        141920   .031  .5126
     DNA for block4.rng      using bits 11 to 12        142227   .937  .8256
     DNA for block4.rng      using bits 10 to 11        142588  2.002  .9774
     DNA for block4.rng      using bits  9 to 10        141927   .052  .5208
     DNA for block4.rng      using bits  8 to  9        141905  -.013  .4949
     DNA for block4.rng      using bits  7 to  8        141491 -1.234  .1086
     DNA for block4.rng      using bits  6 to  7        142005   .282  .6111
     DNA for block4.rng      using bits  5 to  6        142228   .940  .8264
     DNA for block4.rng      using bits  4 to  5        141654  -.753  .2257
     DNA for block4.rng      using bits  3 to  4        142163   .748  .7729
     DNA for block4.rng      using bits  2 to  3        142142   .686  .7538
     DNA for block4.rng      using bits  1 to  2        142035   .371  .6446

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
     ::     This is the COUNT-THE-1's TEST on a stream of bytes.      ::
     :: Consider the file under test as a stream of bytes (four per   ::
     :: 32 bit integer).  Each byte can contain from 0 to 8 1's,      ::
     :: with probabilities 1,8,28,56,70,56,28,8,1 over 256.  Now let  ::
     :: the stream of bytes provide a string of overlapping  5-letter ::
     :: words, each "letter" taking values A,B,C,D,E. The letters are ::
     :: determined by the number of 1's in a byte::  0,1,or 2 yield A,::
     :: 3 yields B, 4 yields C, 5 yields D and 6,7 or 8 yield E. Thus ::
     :: we have a monkey at a typewriter hitting five keys with vari- ::
     :: ous probabilities (37,56,70,56,37 over 256).  There are 5^5   ::
     :: possible 5-letter words, and from a string of 256,000 (over-  ::
     :: lapping) 5-letter words, counts are made on the frequencies   ::
     :: for each word.   The quadratic form in the weak inverse of    ::
     :: the covariance matrix of the cell counts provides a chisquare ::
     :: test::  Q5-Q4, the difference of the naive Pearson sums of    ::
     :: (OBS-EXP)^2/EXP on counts for 5- and 4-letter cell counts.    ::
     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
   Test results for block4.rng
 Chi-square with 5^5-5^4=2500 d.of f. for sample size:2560000
                              chisquare  equiv normal  p-value
  Results fo COUNT-THE-1's in successive bytes:
 byte stream for block4.rng       2448.79      -.724      .234453
 byte stream for block4.rng       2432.88      -.949      .171271

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
     ::     This is the COUNT-THE-1's TEST for specific bytes.        ::
     :: Consider the file under test as a stream of 32-bit integers.  ::
     :: From each integer, a specific byte is chosen , say the left-  ::
     :: most::  bits 1 to 8. Each byte can contain from 0 to 8 1's,   ::
     :: with probabilitie 1,8,28,56,70,56,28,8,1 over 256.  Now let   ::
     :: the specified bytes from successive integers provide a string ::
     :: of (overlapping) 5-letter words, each "letter" taking values  ::
     :: A,B,C,D,E. The letters are determined  by the number of 1's,  ::
     :: in that byte::  0,1,or 2 ---> A, 3 ---> B, 4 ---> C, 5 ---> D,::
     :: and  6,7 or 8 ---> E.  Thus we have a monkey at a typewriter  ::
```

```
:: hitting five keys with with various probabilities::  37,56,70,::
:: 56,37 over 256. There are 5^5 possible 5-letter words, and    ::
:: from a string of 256,000 (overlapping) 5-letter words, counts ::
:: are made on the frequencies for each word. The quadratic form ::
:: in the weak inverse of the covariance matrix of the cell      ::
:: counts provides a chisquare test::  Q5-Q4, the difference of  ::
:: the naive Pearson  sums of (OBS-EXP)^2/EXP on counts for 5-   ::
:: and 4-letter cell counts.                                     ::
:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
```

Chi-square with 5^5-5^4=2500 d.of f. for sample size: 256000

Results for COUNT-THE-1's in specified bytes:

	chisquare	equiv normal	p value
bits 1 to 8	2475.21	-.351	.362958
bits 2 to 9	2482.52	-.247	.402366
bits 3 to 10	2638.34	1.956	.974796
bits 4 to 11	2528.06	.397	.654256
bits 5 to 12	2473.52	-.375	.354010
bits 6 to 13	2475.82	-.342	.366174
bits 7 to 14	2492.27	-.109	.456466
bits 8 to 15	2502.12	.030	.511973
bits 9 to 16	2561.13	.865	.806365
bits 10 to 17	2496.56	-.049	.480585
bits 11 to 18	2471.19	-.408	.341818
bits 12 to 19	2496.23	-.053	.478756
bits 13 to 20	2499.48	-.007	.497068
bits 14 to 21	2439.40	-.857	.195723
bits 15 to 22	2468.71	-.443	.329055
bits 16 to 23	2469.37	-.433	.332431
bits 17 to 24	2517.06	.241	.595329
bits 18 to 25	2556.00	.792	.785821
bits 19 to 26	2502.17	.031	.512268
bits 20 to 27	2439.86	-.851	.197505
bits 21 to 28	2523.74	.336	.631465
bits 22 to 29	2594.99	1.343	.910422
bits 23 to 30	2466.27	-.477	.316674
bits 24 to 31	2513.49	.191	.575654
bits 25 to 32	2519.88	.281	.610725

$$$

```
:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
::               THIS IS A PARKING LOT TEST                      ::
:: In a square of side 100, randomly "park" a car---a circle of  ::
:: radius 1.   Then try to park a 2nd, a 3rd, and so on, each    ::
:: time parking "by ear".  That is, if an attempt to park a car  ::
:: causes a crash with one already parked, try again at a new    ::
:: random location. (To avoid path problems, consider parking    ::
:: helicopters rather than cars.)   Each attempt leads to either ::
:: a crash or a success, the latter followed by an increment to  ::
:: the list of cars already parked. If we plot n:  the number of ::
:: attempts, versus k::  the number successfully parked, we get a::
:: curve that should be similar to those provided by a perfect   ::
:: random number generator.  Theory for the behavior of such a   ::
:: random curve seems beyond reach, and as graphics displays are ::
:: not available for this battery of tests, a simple characteriz ::
:: ation of the random experiment is used: k, the number of cars ::
:: successfully parked after n=12,000 attempts. Simulation shows ::
```

```
:: that k should average 3523 with sigma 21.9 and is very close  ::
:: to normally distributed.  Thus (k-3523)/21.9 should be a st-  ::
:: andard normal variable, which, converted to a uniform varia-  ::
:: ble, provides input to a KSTEST based on a sample of 10.      ::
:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
      CDPARK: result of ten tests on file block4.rng
       Of 12,000 tries, the average no. of successes
            should be 3523 with sigma=21.9
       Successes: 3508    z-score:  -.685 p-value: .246694
       Successes: 3536    z-score:   .594 p-value: .723613
       Successes: 3532    z-score:   .411 p-value: .659449
       Successes: 3540    z-score:   .776 p-value: .781201
       Successes: 3520    z-score:  -.137 p-value: .445521
       Successes: 3511    z-score:  -.548 p-value: .291865
       Successes: 3470    z-score: -2.420 p-value: .007758
       Successes: 3549    z-score:  1.187 p-value: .882429
       Successes: 3528    z-score:   .228 p-value: .590298
       Successes: 3518    z-score:  -.228 p-value: .409702

      square size   avg. no.  parked   sample sigma
        100.            3521.200       21.018
       KSTEST for the above 10: p=  .108721

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
::               THE MINIMUM DISTANCE TEST                       ::
:: It does this 100 times::   choose n=8000 random points in a   ::
:: square of side 10000.  Find d, the minimum distance between   ::
:: the (n^2-n)/2 pairs of points.  If the points are truly inde- ::
:: pendent uniform, then d^2, the square of the minimum distance ::
:: should be (very close to) exponentially distributed with mean ::
:: .995 .  Thus 1-exp(-d^2/.995) should be uniform on [0,1) and  ::
:: a KSTEST on the resulting 100 values serves as a test of uni- ::
:: formity for random points in the square. Test numbers=0 mod 5 ::
:: are printed but the KSTEST is based on the full set of 100    ::
:: random choices of 8000 points in the 10000x10000 square.      ::
:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
          This is the MINIMUM DISTANCE test
         for random integers in the file block4.rng
Sample no.    d^2     avg     equiv uni
      5    1.6292   1.3676     .805505
     10    3.8078   1.8245     .978223
     15     .6531   1.3694     .481281
     20     .3982   1.1326     .329785
     25     .7857   1.0973     .546014
     30    3.3993   1.2014     .967169
     35     .3990   1.0973     .330366
     40     .1931   1.1490     .176409
     45     .8038   1.0844     .554174
     50     .0588   1.0096     .057397
     55    1.4598    .9823     .769418
     60     .5192    .9575     .406552
     65     .9622   1.0115     .619781
     70    3.6999   1.0399     .975729
     75    2.1256   1.0136     .881904
     80     .0657    .9929     .063905
```

```
         85     .3357   1.0291     .286383
         90    2.8616   1.0488     .943640
         95    1.9800   1.0975     .863294
        100     .1482   1.0511     .138368
     MINIMUM DISTANCE TEST for block4.rng
          Result of KS test on 20 transformed mindist^2's:
                                  p-value= .524298

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

     :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
     ::               THE 3DSPHERES TEST                              ::
     :: Choose  4000 random points in a cube of edge 1000.  At each   ::
     :: point, center a sphere large enough to reach the next closest ::
     :: point. Then the volume of the smallest such sphere is (very   ::
     :: close to) exponentially distributed with mean 120pi/3.  Thus  ::
     :: the radius cubed is exponential with mean 30. (The mean is    ::
     :: obtained by extensive simulation).  The 3DSPHERES test gener- ::
     :: ates 4000 such spheres 20 times.  Each min radius cubed leads ::
     :: to a uniform variable by means of 1-exp(-r^3/30.), then a     ::
     ::  KSTEST is done on the 20 p-values.                           ::
     :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
               The 3DSPHERES test for file block4.rng
 sample no:  1     r^3=  22.859     p-value= .53325
 sample no:  2     r^3=   2.518     p-value= .08052
 sample no:  3     r^3=   2.226     p-value= .07150
 sample no:  4     r^3=  19.656     p-value= .48067
 sample no:  5     r^3=   7.559     p-value= .22274
 sample no:  6     r^3=   1.100     p-value= .03600
 sample no:  7     r^3=   6.911     p-value= .20575
 sample no:  8     r^3=  28.342     p-value= .61122
 sample no:  9     r^3=  22.012     p-value= .51988
 sample no: 10     r^3=  10.309     p-value= .29081
 sample no: 11     r^3=  68.051     p-value= .89652
 sample no: 12     r^3=  16.500     p-value= .42305
 sample no: 13     r^3=  31.768     p-value= .65317
 sample no: 14     r^3=  36.106     p-value= .69987
 sample no: 15     r^3=  20.741     p-value= .49911
 sample no: 16     r^3=   2.906     p-value= .09231
 sample no: 17     r^3=  43.456     p-value= .76509
 sample no: 18     r^3=  16.930     p-value= .43126
 sample no: 19     r^3=   6.718     p-value= .20063
 sample no: 20     r^3=  55.060     p-value= .84044
  A KS test is applied to those 20 p-values.
---------------------------------------------------------
       3DSPHERES test for file block4.rng           p-value= .564876
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

     :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
     ::      This is the SQEEZE test                                  ::
     ::  Random integers are floated to get uniforms on [0,1). Start- ::
     ::  ing with k=2^31=2147483647, the test finds j, the number of  ::
     ::  iterations necessary to reduce k to 1, using the reduction   ::
     ::  k=ceiling(k*U), with U provided by floating integers from    ::
     ::  the file being tested.  Such j's are found 100,000 times,    ::
     ::  then counts for the number of times j was <=6,7,...,47,>=48  ::
     ::  are used to provide a chi-square test for cell frequencies.  ::
```

```
::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
      RESULTS OF SQUEEZE TEST FOR block4.rng
   Table of standardized frequency counts
( (obs-exp)/sqrt(exp) )^2
   for j taking values <=6,7,8,...,47,>=48:
  .6    -1.2      .8      .8     -.1      .3
 -.1      .2     1.4     -.8     -.2    -1.5
-1.6     -.2    -1.3      .8     -.1      .1
 -.5     1.7      .5      .3     1.1     1.2
 -.8     -.7    -1.4      .9     1.6      .0
-1.0    -1.0     1.4      .1      .9     1.5
  .0    -1.0    -1.2     2.6      .1      .0
 1.8
      Chi-square with 42 degrees of freedom: 45.074
         z-score=   .335  p-value= .655516
```

$$

```
::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
::            The  OVERLAPPING SUMS test                        ::
:: Integers are floated to get a sequence U(1),U(2),... of uni-  ::
:: form [0,1) variables.  Then overlapping sums,                ::
::   S(1)=U(1)+...+U(100), S2=U(2)+...+U(101),... are formed.    ::
:: The S's are virtually normal with a certain covariance mat-   ::
:: rix.  A linear transformation of the S's converts them to a   ::
:: sequence of independent standard normals, which are converted ::
:: to uniform variables for a KSTEST. The  p-values from ten     ::
:: KSTESTs are given still another KSTEST.                       ::
::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
           Test no.  1      p-value  .303030
           Test no.  2      p-value  .396281
           Test no.  3      p-value  .628466
           Test no.  4      p-value  .047756
           Test no.  5      p-value  .284294
           Test no.  6      p-value  .225097
           Test no.  7      p-value  .904057
           Test no.  8      p-value  .596582
           Test no.  9      p-value  .050642
           Test no. 10      p-value  .713517
   Results of the OSUM test for block4.rng
        KSTEST on the above 10 p-values:  .390328
```

$$

```
::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
::     This is the RUNS test.  It counts runs up, and runs down, ::
:: in a sequence of uniform [0,1) variables, obtained by float-  ::
:: ing the 32-bit integers in the specified file. This example   ::
:: shows how runs are counted:  .123,.357,.789,.425,.224,.416,.95::
:: contains an up-run of length 3, a down-run of length 2 and an ::
:: up-run of (at least) 2, depending on the next values.  The    ::
:: covariance matrices for the runs-up and runs-down are well    ::
:: known, leading to chisquare tests for quadratic forms in the  ::
:: weak inverses of the covariance matrices.  Runs are counted   ::
:: for sequences of length 10,000.  This is done ten times. Then ::
:: repeated.                                                     ::
```

```
::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
      The RUNS test for file block4.rng
Up and down runs in a sample of 10000
```

```
                Run test for block4.rng     :
  runs up; ks test for 10 p's: .057743
runs down; ks test for 10 p's: .306001
                Run test for block4.rng     :
  runs up; ks test for 10 p's: .999245
runs down; ks test for 10 p's: .668036
```

$$

```
::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
:: This is the CRAPS TEST. It plays 200,000 games of craps, finds::
:: the number of wins and the number of throws necessary to end  ::
:: each game.  The number of wins should be (very close to) a    ::
:: normal with mean 200000p and variance 200000p(1-p), with      ::
:: p=244/495.  Throws necessary to complete the game can vary    ::
:: from 1 to infinity, but counts for all>21 are lumped with 21. ::
:: A chi-square test is made on the no.-of-throws cell counts.   ::
:: Each 32-bit integer from the test file provides the value for ::
:: the throw of a die, by floating to [0,1), multiplying by 6    ::
:: and taking 1 plus the integer part of the result.             ::
::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
              Results of craps test for block4.rng
 No. of wins:  Observed Expected
                               98616    98585.86
                98616= No. of wins, z-score=  .135 pvalue= .55362
  Analysis of Throws-per-Game:
Chisq=  20.73 for 20 degrees of freedom, p=  .58697
```

Throws	Observed	Expected	Chisq	Sum
1	66428	66666.7	.854	.854
2	37432	37654.3	1.313	2.167
3	27005	26954.7	.094	2.261
4	19625	19313.5	5.025	7.286
5	14047	13851.4	2.762	10.048
6	9955	9943.5	.013	10.061
7	7117	7145.0	.110	10.171
8	5119	5139.1	.078	10.249
9	3638	3699.9	1.034	11.284
10	2672	2666.3	.012	11.296
11	1968	1923.3	1.038	12.333
12	1357	1388.7	.725	13.059
13	992	1003.7	.137	13.196
14	765	726.1	2.080	15.275
15	512	525.8	.364	15.639
16	379	381.2	.012	15.651
17	273	276.5	.045	15.697
18	173	200.8	3.856	19.553
19	158	146.0	.989	20.542
20	105	106.2	.014	20.556
21	280	287.1	.176	20.732

```
          SUMMARY  FOR block4.rng
              p-value for no. of wins: .553618
              p-value for throws/game: .586970
```

```
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

 Results of DIEHARD battery of tests sent to file report4.txt



       NOTE: Most of the tests in DIEHARD return a p-value, which
       should be uniform on [0,1) if the input file contains truly
       independent random bits.   Those p-values are obtained by
       p=F(X), where F is the assumed distribution of the sample
       random variable X---often normal. But that assumed F is just
       an asymptotic approximation, for which the fit will be worst
       in the tails. Thus you should not be surprised with
       occasional p-values near 0 or 1, such as .0012 or .9983.
       When a bit stream really FAILS BIG, you will get p's of 0 or
       1 to six or more places.  By all means, do not, as a
       Statistician might, think that a p < .025 or p> .975 means
       that the RNG has "failed the test at the .05 level".  Such
       p's happen among the hundreds that DIEHARD produces, even
       with good RNG's.  So keep in mind that " p happens".
     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
     ::            This is the BIRTHDAY SPACINGS TEST                ::
     :: Choose m birthdays in a year of n days.  List the spacings    ::
     :: between the birthdays.  If j is the number of values that     ::
     :: occur more than once in that list, then j is asymptotically   ::
     :: Poisson distributed with mean m^3/(4n).  Experience shows n   ::
     :: must be quite large, say n>=2^18, for comparing the results   ::
     :: to the Poisson distribution with that mean.  This test uses   ::
     :: n=2^24 and m=2^9,  so that the underlying distribution for j  ::
     :: is taken to be Poisson with lambda=2^27/(2^26)=2.  A sample   ::
     :: of 500 j's is taken, and a chi-square goodness of fit test    ::
     :: provides a p value.  The first test uses bits 1-24 (counting  ::
     :: from the left) from integers in the specified file.          ::
     ::   Then the file is closed and reopened. Next, bits 2-25 are   ::
     :: used to provide birthdays, then 3-26 and so on to bits 9-32.  ::
     :: Each set of bits provides a p-value, and the nine p-values    ::
     :: provide a sample for a KSTEST.                               ::
     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
 BIRTHDAY SPACINGS TEST, M= 512 N=2**24 LAMBDA=  2.0000
           Results for block5.rng
                   For a sample of size 500:     mean
           block5.rng      using bits  1 to 24   2.166
  duplicate       number       number
  spacings       observed     expected
        0          57.       67.668
        1         138.      135.335
        2         121.      135.335
        3          94.       90.224
        4          51.       45.112
        5          25.       18.045
  6 to INF         14.        8.282
 Chisquare with  6 d.o.f. =    10.81 p-value=  .905516
  :::::::::::::::::::::::::::::::::::::::::
                   For a sample of size 500:     mean
           block5.rng      using bits  2 to 25   2.086
  duplicate       number       number
```

```
 spacings       observed     expected
      0           66.       67.668
      1          137.      135.335
      2          120.      135.335
      3           92.       90.224
      4           48.       45.112
      5           26.       18.045
 6 to INF         11.        8.282
Chisquare with  6 d.o.f. =     6.42 p-value=  .622026
 :::::::::::::::::::::::::::::::::::::::::
                  For a sample of size 500:     mean
         block5.rng      using bits  3 to 26   2.064
 duplicate       number       number
 spacings       observed     expected
      0           58.       67.668
      1          141.      135.335
      2          117.      135.335
      3          114.       90.224
      4           45.       45.112
      5           17.       18.045
 6 to INF          8.        8.282
Chisquare with  6 d.o.f. =    10.44 p-value=  .892641
 :::::::::::::::::::::::::::::::::::::::::
                  For a sample of size 500:     mean
         block5.rng      using bits  4 to 27   1.978
 duplicate       number       number
 spacings       observed     expected
      0           68.       67.668
      1          147.      135.335
      2          131.      135.335
      3           82.       90.224
      4           44.       45.112
      5           18.       18.045
 6 to INF         10.        8.282
Chisquare with  6 d.o.f. =     2.28 p-value=  .107709
 :::::::::::::::::::::::::::::::::::::::::
                  For a sample of size 500:     mean
         block5.rng      using bits  5 to 28   1.912
 duplicate       number       number
 spacings       observed     expected
      0           73.       67.668
      1          140.      135.335
      2          137.      135.335
      3           80.       90.224
      4           54.       45.112
      5           11.       18.045
 6 to INF          5.        8.282
Chisquare with  6 d.o.f. =     7.56 p-value=  .728012
 :::::::::::::::::::::::::::::::::::::::::
                  For a sample of size 500:     mean
         block5.rng      using bits  6 to 29   1.910
 duplicate       number       number
 spacings       observed     expected
      0           86.       67.668
      1          125.      135.335
      2          130.      135.335
      3           99.       90.224
```

```
       4          36.       45.112
       5          18.       18.045
  6 to INF         6.        8.282
 Chisquare with  6 d.o.f. =     9.29 p-value=  .842040
  ::::::::::::::::::::::::::::::::::::::::
                  For a sample of size 500:     mean
          block5.rng      using bits  7 to 30   2.026
  duplicate       number       number
  spacings       observed     expected
       0          66.       67.668
       1         136.      135.335
       2         145.      135.335
       3          80.       90.224
       4          43.       45.112
       5          11.       18.045
  6 to INF        19.        8.282
 Chisquare with  6 d.o.f. =    18.61 p-value=  .995131
  ::::::::::::::::::::::::::::::::::::::::
                  For a sample of size 500:     mean
          block5.rng      using bits  8 to 31   1.992
  duplicate       number       number
  spacings       observed     expected
       0          57.       67.668
       1         143.      135.335
       2         142.      135.335
       3          94.       90.224
       4          36.       45.112
       5          25.       18.045
  6 to INF         3.        8.282
 Chisquare with  6 d.o.f. =    10.49 p-value=  .894594
  ::::::::::::::::::::::::::::::::::::::::
                  For a sample of size 500:     mean
          block5.rng      using bits  9 to 32   1.996
  duplicate       number       number
  spacings       observed     expected
       0          56.       67.668
       1         146.      135.335
       2         137.      135.335
       3          95.       90.224
       4          45.       45.112
       5          14.       18.045
  6 to INF         7.        8.282
 Chisquare with  6 d.o.f. =     4.23 p-value=  .354530
  ::::::::::::::::::::::::::::::::::::::::
   The 9 p-values were
       .905516   .622026   .892641   .107709   .728012
       .842040   .995131   .894594   .354530
  A KSTEST for the 9 p-values yields  .969266

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

     :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
     ::             THE OVERLAPPING 5-PERMUTATION TEST              ::
     :: This is the OPERM5 test.  It looks at a sequence of one mill- ::
     :: ion 32-bit random integers.  Each set of five consecutive     ::
     :: integers can be in one of 120 states, for the 5! possible or- ::
     :: derings of five numbers.  Thus the 5th, 6th, 7th,...numbers   ::
```

```
     :: each provide a state. As many thousands of state transitions  ::
     :: are observed,  cumulative counts are made of the number of    ::
     :: occurences of each state.  Then the quadratic form in the     ::
     :: weak inverse of the 120x120 covariance matrix yields a test   ::
     :: equivalent to the likelihood ratio test that the 120 cell     ::
     :: counts came from the specified (asymptotically) normal dis-   ::
     :: tribution with the specified 120x120 covariance matrix (with  ::
     :: rank 99).  This version uses 1,000,000 integers, twice.       ::
     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
           OPERM5 test for file block5.rng
     For a sample of 1,000,000 consecutive 5-tuples,
 chisquare for 99 degrees of freedom=110.661; p-value= .801085
           OPERM5 test for file block5.rng
     For a sample of 1,000,000 consecutive 5-tuples,
 chisquare for 99 degrees of freedom=105.649; p-value= .694874
     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
     :: This is the BINARY RANK TEST for 31x31 matrices. The leftmost ::
     :: 31 bits of 31 random integers from the test sequence are used ::
     :: to form a 31x31 binary matrix over the field {0,1}. The rank  ::
     :: is determined. That rank can be from 0 to 31, but ranks< 28   ::
     :: are rare, and their counts are pooled with those for rank 28. ::
     :: Ranks are found for 40,000 such random matrices and a chisqua-::
     :: re test is performed on counts for ranks 31,30,29 and <=28.   ::
     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
    Binary rank test for block5.rng
         Rank test for 31x31 binary matrices:
        rows from leftmost 31 bits of each 32-bit integer
      rank   observed  expected (o-e)^2/e  sum
        28       196     211.4  1.124385    1.124
        29      5165    5134.0   .187059    1.311
        30     23016   23103.0   .327972    1.639
        31     11623   11551.5   .442258    2.082
  chisquare= 2.082 for 3 d. of f.; p-value= .518910
--------------------------------------------------------------
     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
     :: This is the BINARY RANK TEST for 32x32 matrices. A random 32x ::
     :: 32 binary matrix is formed, each row a 32-bit random integer. ::
     :: The rank is determined. That rank can be from 0 to 32, ranks  ::
     :: less than 29 are rare, and their counts are pooled with those ::
     :: for rank 29.  Ranks are found for 40,000 such random matrices ::
     :: and a chisquare test is performed on counts for ranks  32,31, ::
     :: 30 and <=29.                                                   ::
     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
    Binary rank test for block5.rng
         Rank test for 32x32 binary matrices:
        rows from leftmost 32 bits of each 32-bit integer
      rank   observed  expected (o-e)^2/e  sum
        29       204     211.4   .260276     .260
        30      5195    5134.0   .724531     .985
        31     23136   23103.0   .047003    1.032
        32     11465   11551.5   .648094    1.680
  chisquare= 1.680 for 3 d. of f.; p-value= .456058
--------------------------------------------------------------

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
```

```
:: This is the BINARY RANK TEST for 6x8 matrices.  From each of  ::
:: six random 32-bit integers from the generator under test, a   ::
:: specified byte is chosen, and the resulting six bytes form a  ::
:: 6x8 binary matrix whose rank is determined.  That rank can be ::
:: from 0 to 6, but ranks 0,1,2,3 are rare; their counts are     ::
:: pooled with those for rank 4. Ranks are found for 100,000     ::
:: random matrices, and a chi-square test is performed on        ::
:: counts for ranks 6,5 and <=4.                                 ::
:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
    Binary Rank Test for block5.rng
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block5.rng
b-rank test for bits  1 to  8
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4         962        944.3        .332         .332
     r =5        21694     21743.9        .115         .446
     r =6        77344     77311.8        .013         .460
                   p=1-exp(-SUM/2)= .20533
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block5.rng
b-rank test for bits  2 to  9
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4         953        944.3        .080         .080
     r =5        21949     21743.9       1.935        2.015
     r =6        77098     77311.8        .591        2.606
                   p=1-exp(-SUM/2)= .72829
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block5.rng
b-rank test for bits  3 to 10
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4         912        944.3       1.105        1.105
     r =5        21553     21743.9       1.676        2.781
     r =6        77535     77311.8        .644        3.425
                   p=1-exp(-SUM/2)= .81961
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block5.rng
b-rank test for bits  4 to 11
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4         921        944.3        .575         .575
     r =5        21655     21743.9        .363         .938
     r =6        77424     77311.8        .163        1.101
                   p=1-exp(-SUM/2)= .42341
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block5.rng
b-rank test for bits  5 to 12
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4         894        944.3       2.679        2.679
     r =5        21551     21743.9       1.711        4.391
     r =6        77555     77311.8        .765        5.156
                   p=1-exp(-SUM/2)= .92407
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block5.rng
b-rank test for bits  6 to 13
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4         927        944.3        .317         .317
     r =5        21643     21743.9        .468         .785
     r =6        77430     77311.8        .181         .966
```

```
                   p=1-exp(-SUM/2)= .38304
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block5.rng
b-rank test for bits  7 to 14
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          926       944.3         .355         .355
     r =5        21609     21743.9         .837        1.192
     r =6        77465     77311.8         .304        1.495
                   p=1-exp(-SUM/2)= .52649
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block5.rng
b-rank test for bits  8 to 15
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          976       944.3        1.064        1.064
     r =5        21939     21743.9        1.751        2.815
     r =6        77085     77311.8         .665        3.480
                   p=1-exp(-SUM/2)= .82448
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block5.rng
b-rank test for bits  9 to 16
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          929       944.3         .248         .248
     r =5        21718     21743.9         .031         .279
     r =6        77353     77311.8         .022         .301
                   p=1-exp(-SUM/2)= .13961
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block5.rng
b-rank test for bits 10 to 17
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          943       944.3         .002         .002
     r =5        21843     21743.9         .452         .453
     r =6        77214     77311.8         .124         .577
                   p=1-exp(-SUM/2)= .25068
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block5.rng
b-rank test for bits 11 to 18
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          955       944.3         .121         .121
     r =5        21837     21743.9         .399         .520
     r =6        77208     77311.8         .139         .659
                   p=1-exp(-SUM/2)= .28079
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block5.rng
b-rank test for bits 12 to 19
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          918       944.3         .733         .733
     r =5        21864     21743.9         .663        1.396
     r =6        77218     77311.8         .114        1.510
                   p=1-exp(-SUM/2)= .52993
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block5.rng
b-rank test for bits 13 to 20
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          926       944.3         .355         .355
     r =5        21828     21743.9         .325         .680
     r =6        77246     77311.8         .056         .736
                   p=1-exp(-SUM/2)= .30788
```

```
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block5.rng
b-rank test for bits 14 to 21
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          954       944.3         .100         .100
     r =5        21781     21743.9         .063         .163
     r =6        77265     77311.8         .028         .191
                   p=1-exp(-SUM/2)= .09119
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block5.rng
b-rank test for bits 15 to 22
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          985       944.3        1.754        1.754
     r =5        21660     21743.9         .324        2.078
     r =6        77355     77311.8         .024        2.102
                   p=1-exp(-SUM/2)= .65040
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block5.rng
b-rank test for bits 16 to 23
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          958       944.3         .199         .199
     r =5        21737     21743.9         .002         .201
     r =6        77305     77311.8         .001         .202
                   p=1-exp(-SUM/2)= .09585
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block5.rng
b-rank test for bits 17 to 24
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          941       944.3         .012         .012
     r =5        21786     21743.9         .082         .093
     r =6        77273     77311.8         .019         .113
                   p=1-exp(-SUM/2)= .05471
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block5.rng
b-rank test for bits 18 to 25
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          998       944.3        3.054        3.054
     r =5        21837     21743.9         .399        3.452
     r =6        77165     77311.8         .279        3.731
                   p=1-exp(-SUM/2)= .84518
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block5.rng
b-rank test for bits 19 to 26
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          955       944.3         .121         .121
     r =5        21906     21743.9        1.208        1.330
     r =6        77139     77311.8         .386        1.716
                   p=1-exp(-SUM/2)= .57597
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block5.rng
b-rank test for bits 20 to 27
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4         1003       944.3        3.649        3.649
     r =5        21713     21743.9         .044        3.693
     r =6        77284     77311.8         .010        3.703
                   p=1-exp(-SUM/2)= .84297
   Rank of a 6x8 binary matrix,
```

```
     rows formed from eight bits of the RNG block5.rng
     b-rank test for bits 21 to 28
                     OBSERVED   EXPECTED     (O-E)^2/E      SUM
          r<=4          980       944.3       1.350        1.350
          r =5        21684     21743.9        .165        1.515
          r =6        77336     77311.8        .008        1.522
                       p=1-exp(-SUM/2)= .53284
        Rank of a 6x8 binary matrix,
     rows formed from eight bits of the RNG block5.rng
     b-rank test for bits 22 to 29
                     OBSERVED   EXPECTED     (O-E)^2/E      SUM
          r<=4          950       944.3        .034         .034
          r =5        21782     21743.9        .067         .101
          r =6        77268     77311.8        .025         .126
                       p=1-exp(-SUM/2)= .06104
        Rank of a 6x8 binary matrix,
     rows formed from eight bits of the RNG block5.rng
     b-rank test for bits 23 to 30
                     OBSERVED   EXPECTED     (O-E)^2/E      SUM
          r<=4          954       944.3        .100         .100
          r =5        21868     21743.9        .708         .808
          r =6        77178     77311.8        .232        1.039
                       p=1-exp(-SUM/2)= .40532
        Rank of a 6x8 binary matrix,
     rows formed from eight bits of the RNG block5.rng
     b-rank test for bits 24 to 31
                     OBSERVED   EXPECTED     (O-E)^2/E      SUM
          r<=4          998       944.3       3.054        3.054
          r =5        21666     21743.9        .279        3.333
          r =6        77336     77311.8        .008        3.340
                       p=1-exp(-SUM/2)= .81178
        Rank of a 6x8 binary matrix,
     rows formed from eight bits of the RNG block5.rng
     b-rank test for bits 25 to 32
                     OBSERVED   EXPECTED     (O-E)^2/E      SUM
          r<=4          937       944.3        .056         .056
          r =5        21604     21743.9        .900         .957
          r =6        77459     77311.8        .280        1.237
                       p=1-exp(-SUM/2)= .46120
   TEST SUMMARY, 25 tests on 100,000 random 6x8 matrices
 These should be 25 uniform [0,1] random variables:
     .205326     .728286     .819611     .423415     .924066
     .383041     .526494     .824479     .139611     .250680
     .280792     .529926     .307875     .091195     .650404
     .095847     .054711     .845183     .575971     .842974
     .532837     .061042     .405322     .811781     .461200
   brank test summary for block5.rng
       The KS test for those 25 supposed UNI's yields
                    KS p-value= .147032

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
     ::                   THE BITSTREAM TEST                         ::
     :: The file under test is viewed as a stream of bits. Call them  ::
     :: b1,b2,... .  Consider an alphabet with two "letters", 0 and 1 ::
     :: and think of the stream of bits as a succession of 20-letter  ::
```

```
     :: "words", overlapping.  Thus the first word is b1b2...b20, the ::
     :: second is b2b3...b21, and so on.  The bitstream test counts   ::
     :: the number of missing 20-letter (20-bit) words in a string of ::
     :: 2^21 overlapping 20-letter words.  There are 2^20 possible 20 ::
     :: letter words.  For a truly random string of 2^21+19 bits, the ::
     :: number of missing words j should be (very close to) normally  ::
     :: distributed with mean 141,909 and sigma 428.  Thus            ::
     ::  (j-141909)/428 should be a standard normal variate (z score) ::
     :: that leads to a uniform [0,1) p value.  The test is repeated  ::
     :: twenty times.                                                 ::
     :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
 THE OVERLAPPING 20-tuples BITSTREAM  TEST, 20 BITS PER WORD, N words
   This test uses N=2^21 and samples the bitstream 20 times.
  No. missing words should average  141909. with sigma=428.
---------------------------------------------------------
 tst no  1:  142436 missing words,    1.23 sigmas from mean, p-value= .89075
 tst no  2:  141476 missing words,   -1.01 sigmas from mean, p-value= .15566
 tst no  3:  141961 missing words,     .12 sigmas from mean, p-value= .54805
 tst no  4:  141534 missing words,    -.88 sigmas from mean, p-value= .19026
 tst no  5:  141862 missing words,    -.11 sigmas from mean, p-value= .45597
 tst no  6:  141509 missing words,    -.94 sigmas from mean, p-value= .17480
 tst no  7:  141257 missing words,   -1.52 sigmas from mean, p-value= .06374
 tst no  8:  141885 missing words,    -.06 sigmas from mean, p-value= .47734
 tst no  9:  141996 missing words,     .20 sigmas from mean, p-value= .58024
 tst no 10:  142092 missing words,     .43 sigmas from mean, p-value= .66524
 tst no 11:  142101 missing words,     .45 sigmas from mean, p-value= .67286
 tst no 12:  141722 missing words,    -.44 sigmas from mean, p-value= .33081
 tst no 13:  141687 missing words,    -.52 sigmas from mean, p-value= .30172
 tst no 14:  141570 missing words,    -.79 sigmas from mean, p-value= .21394
 tst no 15:  142299 missing words,     .91 sigmas from mean, p-value= .81871
 tst no 16:  142034 missing words,     .29 sigmas from mean, p-value= .61458
 tst no 17:  141515 missing words,    -.92 sigmas from mean, p-value= .17844
 tst no 18:  142247 missing words,     .79 sigmas from mean, p-value= .78493
 tst no 19:  142131 missing words,     .52 sigmas from mean, p-value= .69774
 tst no 20:  141383 missing words,   -1.23 sigmas from mean, p-value= .10940

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

     :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
     ::             The tests OPSO, OQSO and DNA                      ::
     ::         OPSO means Overlapping-Pairs-Sparse-Occupancy         ::
     :: The OPSO test considers 2-letter words from an alphabet of    ::
     :: 1024 letters.  Each letter is determined by a specified ten   ::
     :: bits from a 32-bit integer in the sequence to be tested. OPSO ::
     :: generates  2^21 (overlapping) 2-letter words  (from 2^21+1    ::
     :: "keystrokes")  and counts the number of missing words---that  ::
     :: is 2-letter words which do not appear in the entire sequence. ::
     :: That count should be very close to normally distributed with  ::
     :: mean 141,909, sigma 290. Thus (missingwrds-141909)/290 should ::
     :: be a standard normal variable. The OPSO test takes 32 bits at ::
     :: a time from the test file and uses a designated set of ten    ::
     :: consecutive bits. It then restarts the file for the next de-  ::
     :: signated 10 bits, and so on.                                  ::
     ::                                                               ::
     ::     OQSO means Overlapping-Quadruples-Sparse-Occupancy        ::
     ::   The test OQSO is similar, except that it considers 4-letter ::
     :: words from an alphabet of 32 letters, each letter determined  ::
```

```
    :: by a designated string of 5 consecutive bits from the test    ::
    :: file, elements of which are assumed 32-bit random integers.   ::
    :: The mean number of missing words in a sequence of 2^21 four-  ::
    :: letter words,  (2^21+3 "keystrokes"), is again 141909, with   ::
    :: sigma = 295.  The mean is based on theory; sigma comes from   ::
    :: extensive simulation.                                         ::
    ::                                                               ::
    ::    The DNA test considers an alphabet of 4 letters::  C,G,A,T,::
    :: determined by two designated bits in the sequence of random   ::
    :: integers being tested.  It considers 10-letter words, so that ::
    :: as in OPSO and OQSO, there are 2^20 possible words, and the   ::
    :: mean number of missing words from a string of 2^21  (over-    ::
    :: lapping)  10-letter  words (2^21+9 "keystrokes") is 141909.   ::
    :: The standard deviation sigma=339 was determined as for OQSO   ::
    :: by simulation.  (Sigma for OPSO, 290, is the true value (to   ::
    :: three places), not determined by simulation.                  ::
    :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
OPSO test for generator block5.rng
 Output: No. missing words (mw), equiv normal variate (z), p-value (p)
                                                           mw     z     p
   OPSO for block5.rng      using bits 23 to 32       141425 -1.670  .0475
   OPSO for block5.rng      using bits 22 to 31       141973   .220  .5869
   OPSO for block5.rng      using bits 21 to 30       141694  -.743  .2289
   OPSO for block5.rng      using bits 20 to 29       142800  3.071  .9989
   OPSO for block5.rng      using bits 19 to 28       141799  -.380  .3518
   OPSO for block5.rng      using bits 18 to 27       142180   .933  .8247
   OPSO for block5.rng      using bits 17 to 26       142165   .882  .8110
   OPSO for block5.rng      using bits 16 to 25       141407 -1.732  .0416
   OPSO for block5.rng      using bits 15 to 24       141704  -.708  .2395
   OPSO for block5.rng      using bits 14 to 23       141730  -.618  .2682
   OPSO for block5.rng      using bits 13 to 22       141674  -.811  .2085
   OPSO for block5.rng      using bits 12 to 21       142063   .530  .7019
   OPSO for block5.rng      using bits 11 to 20       141980   .244  .5963
   OPSO for block5.rng      using bits 10 to 19       142301  1.351  .9116
   OPSO for block5.rng      using bits  9 to 18       141937   .095  .5380
   OPSO for block5.rng      using bits  8 to 17       142101   .661  .7457
   OPSO for block5.rng      using bits  7 to 16       141883  -.091  .4638
   OPSO for block5.rng      using bits  6 to 15       141656  -.874  .1912
   OPSO for block5.rng      using bits  5 to 14       142344  1.499  .9330
   OPSO for block5.rng      using bits  4 to 13       142220  1.071  .8580
   OPSO for block5.rng      using bits  3 to 12       141936   .092  .5366
   OPSO for block5.rng      using bits  2 to 11       142459  1.895  .9710
   OPSO for block5.rng      using bits  1 to 10       141754  -.536  .2961
OQSO test for generator block5.rng
 Output: No. missing words (mw), equiv normal variate (z), p-value (p)
                                                           mw     z     p
   OQSO for block5.rng      using bits 28 to 32       141762  -.499  .3087
   OQSO for block5.rng      using bits 27 to 31       141266 -2.181  .0146
   OQSO for block5.rng      using bits 26 to 30       141987   .263  .6038
   OQSO for block5.rng      using bits 25 to 29       141623  -.971  .1659
   OQSO for block5.rng      using bits 24 to 28       141769  -.476  .3171
   OQSO for block5.rng      using bits 23 to 27       141927   .060  .5239
   OQSO for block5.rng      using bits 22 to 26       141659  -.849  .1981
   OQSO for block5.rng      using bits 21 to 25       141851  -.198  .4216
   OQSO for block5.rng      using bits 20 to 24       142038   .436  .6686
   OQSO for block5.rng      using bits 19 to 23       142035   .426  .6649
   OQSO for block5.rng      using bits 18 to 22       141723  -.632  .2638
```

```
    OQSO for block5.rng      using bits 17 to 21        141911   .006  .5023
    OQSO for block5.rng      using bits 16 to 20        142173   .894  .8143
    OQSO for block5.rng      using bits 15 to 19        142042   .450  .6735
    OQSO for block5.rng      using bits 14 to 18        141463 -1.513  .0651
    OQSO for block5.rng      using bits 13 to 17        141657  -.855  .1962
    OQSO for block5.rng      using bits 12 to 16        141961   .175  .5695
    OQSO for block5.rng      using bits 11 to 15        141732  -.601  .2739
    OQSO for block5.rng      using bits 10 to 14        142475  1.918  .9724
    OQSO for block5.rng      using bits  9 to 13        142126   .734  .7687
    OQSO for block5.rng      using bits  8 to 12        141594 -1.069  .1426
    OQSO for block5.rng      using bits  7 to 11        141861  -.164  .4349
    OQSO for block5.rng      using bits  6 to 10        141521 -1.316  .0940
    OQSO for block5.rng      using bits  5 to  9        141727  -.618  .2683
    OQSO for block5.rng      using bits  4 to  8        141671  -.808  .2096
    OQSO for block5.rng      using bits  3 to  7        141968   .199  .5788
    OQSO for block5.rng      using bits  2 to  6        142115   .697  .7572
    OQSO for block5.rng      using bits  1 to  5        142156   .836  .7985
  DNA test for generator block5.rng
  Output: No. missing words (mw), equiv normal variate (z), p-value (p)
                                                           mw     z     p
     DNA for block5.rng      using bits 31 to 32        142050   .415  .6609
     DNA for block5.rng      using bits 30 to 31        141928   .055  .5220
     DNA for block5.rng      using bits 29 to 30        141991   .241  .5952
     DNA for block5.rng      using bits 28 to 29        141970   .179  .5710
     DNA for block5.rng      using bits 27 to 28        142266  1.052  .8536
     DNA for block5.rng      using bits 26 to 27        142099   .560  .7121
     DNA for block5.rng      using bits 25 to 26        142134   .663  .7463
     DNA for block5.rng      using bits 24 to 25        141095 -2.402  .0081
     DNA for block5.rng      using bits 23 to 24        141670  -.706  .2401
     DNA for block5.rng      using bits 22 to 23        141910   .002  .5008
     DNA for block5.rng      using bits 21 to 22        141627  -.833  .2025
     DNA for block5.rng      using bits 20 to 21        142263  1.043  .8516
     DNA for block5.rng      using bits 19 to 20        142075   .489  .6875
     DNA for block5.rng      using bits 18 to 19        142109   .589  .7221
     DNA for block5.rng      using bits 17 to 18        141791  -.349  .3635
     DNA for block5.rng      using bits 16 to 17        142231   .949  .8287
     DNA for block5.rng      using bits 15 to 16        141815  -.278  .3904
     DNA for block5.rng      using bits 14 to 15        142512  1.778  .9623
     DNA for block5.rng      using bits 13 to 14        141999   .265  .6043
     DNA for block5.rng      using bits 12 to 13        142209   .884  .8116
     DNA for block5.rng      using bits 11 to 12        142313  1.191  .8831
     DNA for block5.rng      using bits 10 to 11        141700  -.617  .2685
     DNA for block5.rng      using bits  9 to 10        141994   .250  .5986
     DNA for block5.rng      using bits  8 to  9        142276  1.082  .8603
     DNA for block5.rng      using bits  7 to  8        141998   .262  .6032
     DNA for block5.rng      using bits  6 to  7        142673  2.253  .9879
     DNA for block5.rng      using bits  5 to  6        141662  -.730  .2328
     DNA for block5.rng      using bits  4 to  5        141565 -1.016  .1549
     DNA for block5.rng      using bits  3 to  4        142021   .329  .6291
     DNA for block5.rng      using bits  2 to  3        142102   .568  .7151
     DNA for block5.rng      using bits  1 to  2        142644  2.167  .9849

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
     ::     This is the COUNT-THE-1's TEST on a stream of bytes.     ::
     :: Consider the file under test as a stream of bytes (four per   ::
```

```
    :: 32 bit integer).  Each byte can contain from 0 to 8 1's,      ::
    :: with probabilities 1,8,28,56,70,56,28,8,1 over 256.  Now let  ::
    :: the stream of bytes provide a string of overlapping  5-letter ::
    :: words, each "letter" taking values A,B,C,D,E. The letters are ::
    :: determined by the number of 1's in a byte::  0,1,or 2 yield A,::
    :: 3 yields B, 4 yields C, 5 yields D and 6,7 or 8 yield E. Thus ::
    :: we have a monkey at a typewriter hitting five keys with vari- ::
    :: ous probabilities (37,56,70,56,37 over 256).  There are 5^5   ::
    :: possible 5-letter words, and from a string of 256,000 (over-  ::
    :: lapping) 5-letter words, counts are made on the frequencies   ::
    :: for each word.   The quadratic form in the weak inverse of    ::
    :: the covariance matrix of the cell counts provides a chisquare ::
    :: test::  Q5-Q4, the difference of the naive Pearson sums of    ::
    :: (OBS-EXP)^2/EXP on counts for 5- and 4-letter cell counts.    ::
    ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
  Test results for block5.rng
Chi-square with 5^5-5^4=2500 d.of f. for sample size:2560000
                              chisquare  equiv normal  p-value
 Results fo COUNT-THE-1's in successive bytes:
byte stream for block5.rng       2455.34      -.632      .263829
byte stream for block5.rng       2518.12       .256      .601148

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

    ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
    ::     This is the COUNT-THE-1's TEST for specific bytes.        ::
    :: Consider the file under test as a stream of 32-bit integers.  ::
    :: From each integer, a specific byte is chosen , say the left-  ::
    :: most::  bits 1 to 8. Each byte can contain from 0 to 8 1's,   ::
    :: with probabilitie 1,8,28,56,70,56,28,8,1 over 256.  Now let   ::
    :: the specified bytes from successive integers provide a string ::
    :: of (overlapping) 5-letter words, each "letter" taking values  ::
    :: A,B,C,D,E. The letters are determined  by the number of 1's,  ::
    :: in that byte::  0,1,or 2 ---> A, 3 ---> B, 4 ---> C, 5 ---> D,::
    :: and  6,7 or 8 ---> E.  Thus we have a monkey at a typewriter  ::
    :: hitting five keys with with various probabilities::  37,56,70,::
    :: 56,37 over 256. There are 5^5 possible 5-letter words, and    ::
    :: from a string of 256,000 (overlapping) 5-letter words, counts ::
    :: are made on the frequencies for each word. The quadratic form ::
    :: in the weak inverse of the covariance matrix of the cell      ::
    :: counts provides a chisquare test::  Q5-Q4, the difference of  ::
    :: the naive Pearson  sums of (OBS-EXP)^2/EXP on counts for 5-   ::
    :: and 4-letter cell counts.                                     ::
    ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
Chi-square with 5^5-5^4=2500 d.of f. for sample size: 256000
                      chisquare  equiv normal  p value
 Results for COUNT-THE-1's in specified bytes:
          bits  1 to  8  2428.60     -1.010      .156311
          bits  2 to  9  2496.99      -.043      .483023
          bits  3 to 10  2600.20      1.417      .921759
          bits  4 to 11  2407.92     -1.302      .096415
          bits  5 to 12  2553.04       .750      .773386
          bits  6 to 13  2421.41     -1.111      .133179
          bits  7 to 14  2543.48       .615      .730671
          bits  8 to 15  2587.63      1.239      .892382
          bits  9 to 16  2577.43      1.095      .863243
          bits 10 to 17  2456.08      -.621      .267258
```

```
bits 11 to 18  2544.62       .631      .736000
bits 12 to 19  2493.67      -.090      .464307
bits 13 to 20  2590.92      1.286      .900748
bits 14 to 21  2584.69      1.198      .884494
bits 15 to 22  2550.21       .710      .761156
bits 16 to 23  2503.84       .054      .521670
bits 17 to 24  2444.12      -.790      .214680
bits 18 to 25  2382.54     -1.661      .048342
bits 19 to 26  2454.89      -.638      .261760
bits 20 to 27  2492.49      -.106      .457704
bits 21 to 28  2586.25      1.220      .888716
bits 22 to 29  2434.63      -.925      .177607
bits 23 to 30  2557.27       .810      .790996
bits 24 to 31  2433.07      -.947      .171937
bits 25 to 32  2484.27      -.222      .411975
```

```
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
::              THIS IS A PARKING LOT TEST                      ::
:: In a square of side 100, randomly "park" a car---a circle of  ::
:: radius 1.   Then try to park a 2nd, a 3rd, and so on, each    ::
:: time parking "by ear".  That is, if an attempt to park a car  ::
:: causes a crash with one already parked, try again at a new    ::
:: random location. (To avoid path problems, consider parking    ::
:: helicopters rather than cars.)   Each attempt leads to either ::
:: a crash or a success, the latter followed by an increment to  ::
:: the list of cars already parked. If we plot n:  the number of ::
:: attempts, versus k::  the number successfully parked, we get a::
:: curve that should be similar to those provided by a perfect   ::
:: random number generator.  Theory for the behavior of such a   ::
:: random curve seems beyond reach, and as graphics displays are ::
:: not available for this battery of tests, a simple characteriz ::
:: ation of the random experiment is used: k, the number of cars ::
:: successfully parked after n=12,000 attempts. Simulation shows ::
:: that k should average 3523 with sigma 21.9 and is very close  ::
:: to normally distributed.  Thus (k-3523)/21.9 should be a st-  ::
:: andard normal variable, which, converted to a uniform varia-  ::
:: ble, provides input to a KSTEST based on a sample of 10.      ::
::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
      CDPARK: result of ten tests on file block5.rng
       Of 12,000 tries, the average no. of successes
            should be 3523 with sigma=21.9
       Successes: 3535    z-score:   .548 p-value: .708135
       Successes: 3517    z-score:  -.274 p-value: .392053
       Successes: 3511    z-score:  -.548 p-value: .291865
       Successes: 3514    z-score:  -.411 p-value: .340551
       Successes: 3539    z-score:   .731 p-value: .767486
       Successes: 3496    z-score: -1.233 p-value: .108811
       Successes: 3538    z-score:   .685 p-value: .753306
       Successes: 3501    z-score: -1.005 p-value: .157553
       Successes: 3555    z-score:  1.461 p-value: .928018
       Successes: 3540    z-score:   .776 p-value: .781201

      square size   avg. no.  parked   sample sigma
        100.            3524.600       18.408
       KSTEST for the above 10: p=  .102151
```

```
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

     :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
     ::                THE MINIMUM DISTANCE TEST                       ::
     :: It does this 100 times::   choose n=8000 random points in a   ::
     :: square of side 10000.  Find d, the minimum distance between   ::
     :: the (n^2-n)/2 pairs of points.  If the points are truly inde- ::
     :: pendent uniform, then d^2, the square of the minimum distance ::
     :: should be (very close to) exponentially distributed with mean ::
     :: .995 .  Thus 1-exp(-d^2/.995) should be uniform on [0,1) and  ::
     :: a KSTEST on the resulting 100 values serves as a test of uni- ::
     :: formity for random points in the square. Test numbers=0 mod 5 ::
     :: are printed but the KSTEST is based on the full set of 100    ::
     :: random choices of 8000 points in the 10000x10000 square.      ::
     :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
               This is the MINIMUM DISTANCE test
              for random integers in the file block5.rng
     Sample no.    d^2     avg     equiv uni
           5    1.3199    .4512     .734608
          10    2.6141    .7362     .927724
          15    1.0100    .8543     .637634
          20    1.4339    .8458     .763332
          25     .9678    .7770     .621937
          30     .3322    .9414     .283847
          35     .7507   1.0243     .529719
          40     .3455    .9960     .293364
          45    2.3645   1.0308     .907115
          50     .1963   1.0072     .179006
          55     .4921    .9980     .390175
          60    2.2287   1.0633     .893533
          65     .8783   1.0328     .586345
          70     .7382   1.0267     .523791
          75     .3841   1.0043     .320257
          80     .0208    .9695     .020717
          85     .4425    .9502     .359021
          90     .6310    .9851     .469650
          95     .1621    .9486     .150338
         100     .2302    .9470     .206559
     MINIMUM DISTANCE TEST for block5.rng
          Result of KS test on 20 transformed mindist^2's:
                                  p-value= .503603

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

     :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
     ::               THE 3DSPHERES TEST                               ::
     :: Choose  4000 random points in a cube of edge 1000.  At each   ::
     :: point, center a sphere large enough to reach the next closest ::
     :: point. Then the volume of the smallest such sphere is (very   ::
     :: close to) exponentially distributed with mean 120pi/3.  Thus  ::
     :: the radius cubed is exponential with mean 30. (The mean is    ::
     :: obtained by extensive simulation).  The 3DSPHERES test gener- ::
     :: ates 4000 such spheres 20 times.  Each min radius cubed leads ::
     :: to a uniform variable by means of 1-exp(-r^3/30.), then a     ::
     ::  KSTEST is done on the 20 p-values.                            ::
     :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
```

```
               The 3DSPHERES test for file block5.rng
 sample no:  1     r^3=  29.363     p-value= .62422
 sample no:  2     r^3=   5.798     p-value= .17574
 sample no:  3     r^3=   6.909     p-value= .20571
 sample no:  4     r^3=  26.439     p-value= .58575
 sample no:  5     r^3= 104.293     p-value= .96908
 sample no:  6     r^3=   5.604     p-value= .17038
 sample no:  7     r^3=  13.282     p-value= .35772
 sample no:  8     r^3=  11.082     p-value= .30885
 sample no:  9     r^3=  25.090     p-value= .56670
 sample no: 10     r^3=   5.849     p-value= .17713
 sample no: 11     r^3=   4.839     p-value= .14897
 sample no: 12     r^3= 174.992     p-value= .99707
 sample no: 13     r^3=  56.236     p-value= .84658
 sample no: 14     r^3=  63.491     p-value= .87953
 sample no: 15     r^3=  34.281     p-value= .68104
 sample no: 16     r^3=   2.749     p-value= .08756
 sample no: 17     r^3= 146.778     p-value= .99250
 sample no: 18     r^3=    .062     p-value= .00206
 sample no: 19     r^3=   7.795     p-value= .22881
 sample no: 20     r^3=  10.852     p-value= .30354
  A KS test is applied to those 20 p-values.
---------------------------------------------------------
       3DSPHERES test for file block5.rng           p-value= .743216
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
     ::       This is the SQEEZE test                                  ::
     ::  Random integers are floated to get uniforms on [0,1). Start-  ::
     ::  ing with k=2^31=2147483647, the test finds j, the number of   ::
     ::  iterations necessary to reduce k to 1, using the reduction    ::
     ::  k=ceiling(k*U), with U provided by floating integers from     ::
     ::  the file being tested.  Such j's are found 100,000 times,     ::
     ::  then counts for the number of times j was <=6,7,...,47,>=48   ::
     ::  are used to provide a chi-square test for cell frequencies.   ::
     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
           RESULTS OF SQUEEZE TEST FOR block5.rng
         Table of standardized frequency counts
     ( (obs-exp)/sqrt(exp) )^2
        for j taking values <=6,7,8,...,47,>=48:
    -1.5     -.3      .1     1.4     1.3    -1.7
      .0      .2     -.8    -1.5     -.2     1.2
     -.4     -.7     -.1     1.2      .2      .6
      .2      .0    -1.4      .2    -1.3      .9
     1.1      .5      .3      .2      .0     -.4
     -.8      .6     -.5      .0     -.5    -2.6
      .0    -1.0      .1     1.0     1.6     3.0
     -.1
           Chi-square with 42 degrees of freedom: 43.969
              z-score=   .215  p-value= .611928
______________________________________________________________

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
     ::             The  OVERLAPPING SUMS test                         ::
     :: Integers are floated to get a sequence U(1),U(2),... of uni-   ::
```

```
     :: form [0,1) variables.  Then overlapping sums,                ::
     ::   S(1)=U(1)+...+U(100), S2=U(2)+...+U(101),... are formed.   ::
     :: The S's are virtually normal with a certain covariance mat-  ::
     :: rix.  A linear transformation of the S's converts them to a  ::
     :: sequence of independent standard normals, which are converted ::
     :: to uniform variables for a KSTEST. The  p-values from ten    ::
     :: KSTESTs are given still another KSTEST.                      ::
     :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
                Test no.  1      p-value  .634387
                Test no.  2      p-value  .661018
                Test no.  3      p-value  .080425
                Test no.  4      p-value  .708840
                Test no.  5      p-value  .344657
                Test no.  6      p-value  .768837
                Test no.  7      p-value  .396214
                Test no.  8      p-value  .358390
                Test no.  9      p-value  .027324
                Test no. 10      p-value  .760155
   Results of the OSUM test for block5.rng
        KSTEST on the above 10 p-values:  .290043

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

     :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
     ::     This is the RUNS test.  It counts runs up, and runs down, ::
     :: in a sequence of uniform [0,1) variables, obtained by float-  ::
     :: ing the 32-bit integers in the specified file. This example   ::
     :: shows how runs are counted:  .123,.357,.789,.425,.224,.416,.95::
     :: contains an up-run of length 3, a down-run of length 2 and an ::
     :: up-run of (at least) 2, depending on the next values.  The    ::
     :: covariance matrices for the runs-up and runs-down are well    ::
     :: known, leading to chisquare tests for quadratic forms in the  ::
     :: weak inverses of the covariance matrices.  Runs are counted   ::
     :: for sequences of length 10,000.  This is done ten times. Then ::
     :: repeated.                                                     ::
     :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
           The RUNS test for file block5.rng
     Up and down runs in a sample of 10000
_________________________________________________
                 Run test for block5.rng     :
       runs up; ks test for 10 p's: .741353
     runs down; ks test for 10 p's: .212547
                 Run test for block5.rng     :
       runs up; ks test for 10 p's: .797036
     runs down; ks test for 10 p's: .551932

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

     :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
     :: This is the CRAPS TEST. It plays 200,000 games of craps, finds::
     :: the number of wins and the number of throws necessary to end  ::
     :: each game.  The number of wins should be (very close to) a    ::
     :: normal with mean 200000p and variance 200000p(1-p), with      ::
     :: p=244/495.  Throws necessary to complete the game can vary    ::
     :: from 1 to infinity, but counts for all>21 are lumped with 21. ::
     :: A chi-square test is made on the no.-of-throws cell counts.   ::
     :: Each 32-bit integer from the test file provides the value for ::
```

:: the throw of a die, by floating to [0,1), multiplying by 6 ::
:: and taking 1 plus the integer part of the result. ::
::

Results of craps test for block5.rng

No. of wins: Observed Expected

98747 98585.86

98747= No. of wins, z-score= .721 pvalue= .76446

Analysis of Throws-per-Game:

Chisq= 15.76 for 20 degrees of freedom, p= .26863

Throws	Observed	Expected	Chisq	Sum
1	66377	66666.7	1.259	1.259
2	37848	37654.3	.996	2.255
3	27018	26954.7	.149	2.403
4	19237	19313.5	.303	2.706
5	13946	13851.4	.646	3.352
6	9874	9943.5	.486	3.838
7	7168	7145.0	.074	3.912
8	5254	5139.1	2.570	6.482
9	3662	3699.9	.388	6.870
10	2654	2666.3	.057	6.927
11	1918	1923.3	.015	6.941
12	1337	1388.7	1.928	8.869
13	1025	1003.7	.451	9.320
14	715	726.1	.171	9.491
15	568	525.8	3.381	12.872
16	384	381.2	.021	12.894
17	271	276.5	.111	13.004
18	206	200.8	.133	13.138
19	151	146.0	.172	13.310
20	118	106.2	1.308	14.617
21	269	287.1	1.143	15.760

SUMMARY FOR block5.rng

p-value for no. of wins: .764458

p-value for throws/game: .268631

$$

Results of DIEHARD battery of tests sent to file report5.txt

NOTE: Most of the tests in DIEHARD return a p-value, which should be uniform on [0,1) if the input file contains truly independent random bits. Those p-values are obtained by p=F(X), where F is the assumed distribution of the sample random variable X---often normal. But that assumed F is just an asymptotic approximation, for which the fit will be worst in the tails. Thus you should not be surprised with occasional p-values near 0 or 1, such as .0012 or .9983. When a bit stream really FAILS BIG, you will get p's of 0 or 1 to six or more places. By all means, do not, as a Statistician might, think that a p < .025 or p> .975 means that the RNG has "failed the test at the .05 level". Such p's happen among the hundreds that DIEHARD produces, even with good RNG's. So keep in mind that " p happens".

::

```
    ::            This is the BIRTHDAY SPACINGS TEST                 ::
    :: Choose m birthdays in a year of n days.  List the spacings    ::
    :: between the birthdays.  If j is the number of values that     ::
    :: occur more than once in that list, then j is asymptotically   ::
    :: Poisson distributed with mean m^3/(4n).  Experience shows n   ::
    :: must be quite large, say n>=2^18, for comparing the results   ::
    :: to the Poisson distribution with that mean.  This test uses   ::
    :: n=2^24 and m=2^9,  so that the underlying distribution for j  ::
    :: is taken to be Poisson with lambda=2^27/(2^26)=2.  A sample   ::
    :: of 500 j's is taken, and a chi-square goodness of fit test    ::
    :: provides a p value.  The first test uses bits 1-24 (counting  ::
    :: from the left) from integers in the specified file.           ::
    ::   Then the file is closed and reopened. Next, bits 2-25 are   ::
    :: used to provide birthdays, then 3-26 and so on to bits 9-32.  ::
    :: Each set of bits provides a p-value, and the nine p-values    ::
    :: provide a sample for a KSTEST.                                ::
    :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
BIRTHDAY SPACINGS TEST, M= 512 N=2**24 LAMBDA=  2.0000
          Results for block6.rng
                  For a sample of size 500:     mean
          block6.rng      using bits  1 to 24   1.916
 duplicate       number       number
 spacings       observed     expected
       0          71.       67.668
       1         145.      135.335
       2         143.      135.335
       3          74.       90.224
       4          41.       45.112
       5          18.       18.045
 6 to INF          8.        8.282
Chisquare with  6 d.o.f. =     4.59 p-value=  .402646
 :::::::::::::::::::::::::::::::::::::::::
                  For a sample of size 500:     mean
          block6.rng      using bits  2 to 25   2.010
 duplicate       number       number
 spacings       observed     expected
       0          65.       67.668
       1         148.      135.335
       2         122.      135.335
       3          83.       90.224
       4          56.       45.112
       5          17.       18.045
 6 to INF          9.        8.282
Chisquare with  6 d.o.f. =     5.93 p-value=  .569311
 :::::::::::::::::::::::::::::::::::::::::
                  For a sample of size 500:     mean
          block6.rng      using bits  3 to 26   1.926
 duplicate       number       number
 spacings       observed     expected
       0          74.       67.668
       1         135.      135.335
       2         137.      135.335
       3          92.       90.224
       4          37.       45.112
       5          21.       18.045
 6 to INF          4.        8.282
Chisquare with  6 d.o.f. =     4.81 p-value=  .430976
```

```
 :::::::::::::::::::::::::::::::::::::::::
                 For a sample of size 500:     mean
         block6.rng      using bits  4 to 27   2.066
 duplicate       number       number
 spacings       observed     expected
       0          64.       67.668
       1         122.      135.335
       2         139.      135.335
       3         101.       90.224
       4          48.       45.112
       5          20.       18.045
 6 to INF          6.        8.282
Chisquare with  6 d.o.f. =     3.92 p-value=  .313129
 :::::::::::::::::::::::::::::::::::::::::
                 For a sample of size 500:     mean
         block6.rng      using bits  5 to 28   1.996
 duplicate       number       number
 spacings       observed     expected
       0          63.       67.668
       1         138.      135.335
       2         143.      135.335
       3          86.       90.224
       4          43.       45.112
       5          18.       18.045
 6 to INF          9.        8.282
Chisquare with  6 d.o.f. =     1.17 p-value=  .021540
 :::::::::::::::::::::::::::::::::::::::::
                 For a sample of size 500:     mean
         block6.rng      using bits  6 to 29   1.978
 duplicate       number       number
 spacings       observed     expected
       0          64.       67.668
       1         139.      135.335
       2         142.      135.335
       3          83.       90.224
       4          52.       45.112
       5          13.       18.045
 6 to INF          7.        8.282
Chisquare with  6 d.o.f. =     3.87 p-value=  .305071
 :::::::::::::::::::::::::::::::::::::::::
                 For a sample of size 500:     mean
         block6.rng      using bits  7 to 30   2.010
 duplicate       number       number
 spacings       observed     expected
       0          71.       67.668
       1         135.      135.335
       2         122.      135.335
       3          98.       90.224
       4          45.       45.112
       5          25.       18.045
 6 to INF          4.        8.282
Chisquare with  6 d.o.f. =     7.04 p-value=  .683215
 :::::::::::::::::::::::::::::::::::::::::
                 For a sample of size 500:     mean
         block6.rng      using bits  8 to 31   2.044
 duplicate       number       number
 spacings       observed     expected
```

```
       0          60.       67.668
       1         145.      135.335
       2         134.      135.335
       3          82.       90.224
       4          45.       45.112
       5          24.       18.045
  6 to INF        10.        8.282
 Chisquare with  6 d.o.f. =     4.64 p-value=  .409776
  :::::::::::::::::::::::::::::::::::::::::
                  For a sample of size 500:     mean
         block6.rng      using bits  9 to 32   1.918
  duplicate       number       number
  spacings       observed     expected
       0          53.       67.668
       1         157.      135.335
       2         148.      135.335
       3          80.       90.224
       4          47.       45.112
       5          12.       18.045
  6 to INF         3.        8.282
 Chisquare with  6 d.o.f. =    14.46 p-value=  .975134
  :::::::::::::::::::::::::::::::::::::::::
   The 9 p-values were
       .402646   .569311   .430976   .313129   .021540
       .305071   .683215   .409776   .975134
  A KSTEST for the 9 p-values yields  .388399

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
     ::            THE OVERLAPPING 5-PERMUTATION TEST                 ::
     :: This is the OPERM5 test.  It looks at a sequence of one mill- ::
     :: ion 32-bit random integers.  Each set of five consecutive     ::
     :: integers can be in one of 120 states, for the 5! possible or- ::
     :: derings of five numbers.  Thus the 5th, 6th, 7th,...numbers   ::
     :: each provide a state. As many thousands of state transitions  ::
     :: are observed,  cumulative counts are made of the number of    ::
     :: occurences of each state.  Then the quadratic form in the     ::
     :: weak inverse of the 120x120 covariance matrix yields a test   ::
     :: equivalent to the likelihood ratio test that the 120 cell     ::
     :: counts came from the specified (asymptotically) normal dis-   ::
     :: tribution with the specified 120x120 covariance matrix (with  ::
     :: rank 99).  This version uses 1,000,000 integers, twice.       ::
     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
           OPERM5 test for file block6.rng
     For a sample of 1,000,000 consecutive 5-tuples,
 chisquare for 99 degrees of freedom=111.188; p-value= .810631
           OPERM5 test for file block6.rng
     For a sample of 1,000,000 consecutive 5-tuples,
 chisquare for 99 degrees of freedom=131.353; p-value= .983563
     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
     :: This is the BINARY RANK TEST for 31x31 matrices. The leftmost ::
     :: 31 bits of 31 random integers from the test sequence are used ::
     :: to form a 31x31 binary matrix over the field {0,1}. The rank  ::
     :: is determined. That rank can be from 0 to 31, but ranks< 28   ::
     :: are rare, and their counts are pooled with those for rank 28. ::
     :: Ranks are found for 40,000 such random matrices and a chisqua-::
```

```
     :: re test is performed on counts for ranks 31,30,29 and <=28.   ::
     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
    Binary rank test for block6.rng
         Rank test for 31x31 binary matrices:
        rows from leftmost 31 bits of each 32-bit integer
      rank   observed  expected (o-e)^2/e  sum
        28       204     211.4   .260276     .260
        29      4988    5134.0  4.152503    4.413
        30     23265   23103.0  1.135297    5.548
        31     11543   11551.5   .006291    5.554
  chisquare= 5.554 for 3 d. of f.; p-value= .873795
--------------------------------------------------------------
     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
     :: This is the BINARY RANK TEST for 32x32 matrices. A random 32x ::
     :: 32 binary matrix is formed, each row a 32-bit random integer. ::
     :: The rank is determined. That rank can be from 0 to 32, ranks  ::
     :: less than 29 are rare, and their counts are pooled with those ::
     :: for rank 29.  Ranks are found for 40,000 such random matrices ::
     :: and a chisquare test is performed on counts for ranks  32,31, ::
     :: 30 and <=29.                                                  ::
     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
    Binary rank test for block6.rng
         Rank test for 32x32 binary matrices:
        rows from leftmost 32 bits of each 32-bit integer
      rank   observed  expected (o-e)^2/e  sum
        29       229     211.4  1.462156    1.462
        30      5204    5134.0   .954140    2.416
        31     23050   23103.0   .121801    2.538
        32     11517   11551.5   .103184    2.641
  chisquare= 2.641 for 3 d. of f.; p-value= .601762
--------------------------------------------------------------

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
     :: This is the BINARY RANK TEST for 6x8 matrices.  From each of  ::
     :: six random 32-bit integers from the generator under test, a   ::
     :: specified byte is chosen, and the resulting six bytes form a  ::
     :: 6x8 binary matrix whose rank is determined.  That rank can be ::
     :: from 0 to 6, but ranks 0,1,2,3 are rare; their counts are     ::
     :: pooled with those for rank 4. Ranks are found for 100,000     ::
     :: random matrices, and a chi-square test is performed on        ::
     :: counts for ranks 6,5 and <=4.                                 ::
     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
         Binary Rank Test for block6.rng
        Rank of a 6x8 binary matrix,
     rows formed from eight bits of the RNG block6.rng
     b-rank test for bits  1 to  8
                     OBSERVED   EXPECTED     (O-E)^2/E      SUM
          r<=4          947       944.3        .008        .008
          r =5        21706     21743.9        .066        .074
          r =6        77347     77311.8        .016        .090
                        p=1-exp(-SUM/2)= .04391
        Rank of a 6x8 binary matrix,
     rows formed from eight bits of the RNG block6.rng
     b-rank test for bits  2 to  9
                     OBSERVED   EXPECTED     (O-E)^2/E      SUM
```

```
     r<=4          975       944.3          .998         .998
     r =5        21581     21743.9         1.220        2.218
     r =6        77444     77311.8          .226        2.444
                p=1-exp(-SUM/2)= .70543
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block6.rng
b-rank test for bits  3 to 10
              OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          963       944.3          .370         .370
     r =5        21804     21743.9          .166         .536
     r =6        77233     77311.8          .080         .617
                p=1-exp(-SUM/2)= .26534
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block6.rng
b-rank test for bits  4 to 11
              OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          901       944.3         1.986        1.986
     r =5        21923     21743.9         1.475        3.461
     r =6        77176     77311.8          .239        3.699
                p=1-exp(-SUM/2)= .84271
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block6.rng
b-rank test for bits  5 to 12
              OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          936       944.3          .073         .073
     r =5        21862     21743.9          .641         .714
     r =6        77202     77311.8          .156         .870
                p=1-exp(-SUM/2)= .35286
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block6.rng
b-rank test for bits  6 to 13
              OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          935       944.3          .092         .092
     r =5        22056     21743.9         4.480        4.571
     r =6        77009     77311.8         1.186        5.757
                p=1-exp(-SUM/2)= .94379
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block6.rng
b-rank test for bits  7 to 14
              OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          964       944.3          .411         .411
     r =5        21824     21743.9          .295         .706
     r =6        77212     77311.8          .129         .835
                p=1-exp(-SUM/2)= .34126
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block6.rng
b-rank test for bits  8 to 15
              OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          944       944.3          .000         .000
     r =5        21679     21743.9          .194         .194
     r =6        77377     77311.8          .055         .249
                p=1-exp(-SUM/2)= .11697
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block6.rng
b-rank test for bits  9 to 16
              OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          898       944.3         2.270        2.270
```

```
     r =5         21801     21743.9         .150       2.420
     r =6         77301     77311.8         .002       2.422
                    p=1-exp(-SUM/2)= .70206
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block6.rng
b-rank test for bits 10 to 17
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4           980       944.3        1.350       1.350
     r =5         21645     21743.9         .450       1.799
     r =6         77375     77311.8         .052       1.851
                    p=1-exp(-SUM/2)= .60368
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block6.rng
b-rank test for bits 11 to 18
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4           953       944.3         .080        .080
     r =5         21719     21743.9         .029        .109
     r =6         77328     77311.8         .003        .112
                    p=1-exp(-SUM/2)= .05448
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block6.rng
b-rank test for bits 12 to 19
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4           906       944.3        1.554       1.554
     r =5         21700     21743.9         .089       1.642
     r =6         77394     77311.8         .087       1.730
                    p=1-exp(-SUM/2)= .57885
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block6.rng
b-rank test for bits 13 to 20
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4           917       944.3         .789        .789
     r =5         21684     21743.9         .165        .954
     r =6         77399     77311.8         .098       1.053
                    p=1-exp(-SUM/2)= .40924
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block6.rng
b-rank test for bits 14 to 21
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4           953       944.3         .080        .080
     r =5         21869     21743.9         .720        .800
     r =6         77178     77311.8         .232       1.031
                    p=1-exp(-SUM/2)= .40293
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block6.rng
b-rank test for bits 15 to 22
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4           958       944.3         .199        .199
     r =5         21807     21743.9         .183        .382
     r =6         77235     77311.8         .076        .458
                    p=1-exp(-SUM/2)= .20473
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block6.rng
b-rank test for bits 16 to 23
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4           919       944.3         .678        .678
     r =5         21708     21743.9         .059        .737
```

```
     r =6        77373     77311.8         .048         .786
                  p=1-exp(-SUM/2)= .32484
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block6.rng
b-rank test for bits 17 to 24
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          975       944.3         .998         .998
     r =5        21565     21743.9        1.472        2.470
     r =6        77460     77311.8         .284        2.754
                  p=1-exp(-SUM/2)= .74766
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block6.rng
b-rank test for bits 18 to 25
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          969       944.3         .646         .646
     r =5        21843     21743.9         .452        1.098
     r =6        77188     77311.8         .198        1.296
                  p=1-exp(-SUM/2)= .47689
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block6.rng
b-rank test for bits 19 to 26
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          934       944.3         .112         .112
     r =5        21814     21743.9         .226         .338
     r =6        77252     77311.8         .046         .385
                  p=1-exp(-SUM/2)= .17495
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block6.rng
b-rank test for bits 20 to 27
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          959       944.3         .229         .229
     r =5        22044     21743.9        4.142        4.371
     r =6        76997     77311.8        1.282        5.652
                  p=1-exp(-SUM/2)= .94077
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block6.rng
b-rank test for bits 21 to 28
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          961       944.3         .295         .295
     r =5        21809     21743.9         .195         .490
     r =6        77230     77311.8         .087         .577
                  p=1-exp(-SUM/2)= .25052
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block6.rng
b-rank test for bits 22 to 29
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          936       944.3         .073         .073
     r =5        21668     21743.9         .265         .338
     r =6        77396     77311.8         .092         .430
                  p=1-exp(-SUM/2)= .19330
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block6.rng
b-rank test for bits 23 to 30
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          942       944.3         .006         .006
     r =5        21541     21743.9        1.893        1.899
     r =6        77517     77311.8         .545        2.444
```

```
                    p=1-exp(-SUM/2)= .70530
        Rank of a 6x8 binary matrix,
     rows formed from eight bits of the RNG block6.rng
     b-rank test for bits 24 to 31
                 OBSERVED   EXPECTED     (O-E)^2/E      SUM
          r<=4      1005       944.3       3.902       3.902
          r =5     21627     21743.9        .628       4.530
          r =6     77368     77311.8        .041       4.571
                    p=1-exp(-SUM/2)= .89828
        Rank of a 6x8 binary matrix,
     rows formed from eight bits of the RNG block6.rng
     b-rank test for bits 25 to 32
                 OBSERVED   EXPECTED     (O-E)^2/E      SUM
          r<=4       933       944.3        .135        .135
          r =5     21842     21743.9        .443        .578
          r =6     77225     77311.8        .097        .675
                    p=1-exp(-SUM/2)= .28656
   TEST SUMMARY, 25 tests on 100,000 random 6x8 matrices
 These should be 25 uniform [0,1] random variables:
     .043905     .705426     .265344     .842712     .352857
     .943789     .341255     .116967     .702058     .603680
     .054479     .578851     .409237     .402932     .204726
     .324845     .747664     .476888     .174953     .940765
     .250522     .193301     .705295     .898276     .286555
   brank test summary for block6.rng
       The KS test for those 25 supposed UNI's yields
                  KS p-value= .199101

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
     ::                   THE BITSTREAM TEST                         ::
     :: The file under test is viewed as a stream of bits. Call them  ::
     :: b1,b2,... .  Consider an alphabet with two "letters", 0 and 1 ::
     :: and think of the stream of bits as a succession of 20-letter  ::
     :: "words", overlapping.  Thus the first word is b1b2...b20, the ::
     :: second is b2b3...b21, and so on.  The bitstream test counts   ::
     :: the number of missing 20-letter (20-bit) words in a string of ::
     :: 2^21 overlapping 20-letter words.  There are 2^20 possible 20 ::
     :: letter words.  For a truly random string of 2^21+19 bits, the ::
     :: number of missing words j should be (very close to) normally  ::
     :: distributed with mean 141,909 and sigma 428.  Thus            ::
     ::  (j-141909)/428 should be a standard normal variate (z score) ::
     :: that leads to a uniform [0,1) p value.  The test is repeated  ::
     :: twenty times.                                                 ::
     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
 THE OVERLAPPING 20-tuples BITSTREAM  TEST, 20 BITS PER WORD, N words
   This test uses N=2^21 and samples the bitstream 20 times.
  No. missing words should average  141909. with sigma=428.
---------------------------------------------------------
 tst no  1:  142200 missing words,     .68 sigmas from mean, p-value= .75148
 tst no  2:  142225 missing words,     .74 sigmas from mean, p-value= .76961
 tst no  3:  141968 missing words,     .14 sigmas from mean, p-value= .55452
 tst no  4:  142316 missing words,     .95 sigmas from mean, p-value= .82899
 tst no  5:  141928 missing words,     .04 sigmas from mean, p-value= .51740
 tst no  6:  141668 missing words,    -.56 sigmas from mean, p-value= .28643
 tst no  7:  141738 missing words,    -.40 sigmas from mean, p-value= .34447
```

```
 tst no  8:  142447 missing words,    1.26 sigmas from mean, p-value= .89549
 tst no  9:  141747 missing words,    -.38 sigmas from mean, p-value= .35224
 tst no 10:  142046 missing words,     .32 sigmas from mean, p-value= .62526
 tst no 11:  141714 missing words,    -.46 sigmas from mean, p-value= .32406
 tst no 12:  141939 missing words,     .07 sigmas from mean, p-value= .52764
 tst no 13:  141390 missing words,   -1.21 sigmas from mean, p-value= .11249
 tst no 14:  141639 missing words,    -.63 sigmas from mean, p-value= .26382
 tst no 15:  142330 missing words,     .98 sigmas from mean, p-value= .83717
 tst no 16:  142102 missing words,     .45 sigmas from mean, p-value= .67371
 tst no 17:  141415 missing words,   -1.15 sigmas from mean, p-value= .12405
 tst no 18:  142447 missing words,    1.26 sigmas from mean, p-value= .89549
 tst no 19:  140808 missing words,   -2.57 sigmas from mean, p-value= .00504
 tst no 20:  141488 missing words,    -.98 sigmas from mean, p-value= .16246

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

     :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
     ::             The tests OPSO, OQSO and DNA                       ::
     ::         OPSO means Overlapping-Pairs-Sparse-Occupancy          ::
     :: The OPSO test considers 2-letter words from an alphabet of     ::
     :: 1024 letters.  Each letter is determined by a specified ten    ::
     :: bits from a 32-bit integer in the sequence to be tested. OPSO ::
     :: generates  2^21 (overlapping) 2-letter words  (from 2^21+1     ::
     :: "keystrokes")  and counts the number of missing words---that  ::
     :: is 2-letter words which do not appear in the entire sequence. ::
     :: That count should be very close to normally distributed with  ::
     :: mean 141,909, sigma 290. Thus (missingwrds-141909)/290 should ::
     :: be a standard normal variable. The OPSO test takes 32 bits at ::
     :: a time from the test file and uses a designated set of ten    ::
     :: consecutive bits. It then restarts the file for the next de-  ::
     :: signated 10 bits, and so on.                                   ::
     ::                                                                ::
     ::     OQSO means Overlapping-Quadruples-Sparse-Occupancy         ::
     ::   The test OQSO is similar, except that it considers 4-letter ::
     :: words from an alphabet of 32 letters, each letter determined  ::
     :: by a designated string of 5 consecutive bits from the test    ::
     :: file, elements of which are assumed 32-bit random integers.   ::
     :: The mean number of missing words in a sequence of 2^21 four-  ::
     :: letter words,  (2^21+3 "keystrokes"), is again 141909, with   ::
     :: sigma = 295.  The mean is based on theory; sigma comes from   ::
     :: extensive simulation.                                          ::
     ::                                                                ::
     ::    The DNA test considers an alphabet of 4 letters::  C,G,A,T,::
     :: determined by two designated bits in the sequence of random   ::
     :: integers being tested.  It considers 10-letter words, so that ::
     :: as in OPSO and OQSO, there are 2^20 possible words, and the   ::
     :: mean number of missing words from a string of 2^21  (over-    ::
     :: lapping)  10-letter  words (2^21+9 "keystrokes") is 141909.   ::
     :: The standard deviation sigma=339 was determined as for OQSO   ::
     :: by simulation.  (Sigma for OPSO, 290, is the true value (to   ::
     :: three places), not determined by simulation.                   ::
     :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
 OPSO test for generator block6.rng
  Output: No. missing words (mw), equiv normal variate (z), p-value (p)
                                                           mw     z     p
   OPSO for block6.rng      using bits 23 to 32        141579 -1.139  .1273
   OPSO for block6.rng      using bits 22 to 31        142028   .409  .6588
```

```
   OPSO for block6.rng      using bits 21 to 30       141593 -1.091  .1377
   OPSO for block6.rng      using bits 20 to 29       141516 -1.356  .0875
   OPSO for block6.rng      using bits 19 to 28       142213  1.047  .8525
   OPSO for block6.rng      using bits 18 to 27       142043   .461  .6776
   OPSO for block6.rng      using bits 17 to 26       141772  -.474  .3179
   OPSO for block6.rng      using bits 16 to 25       141778  -.453  .3253
   OPSO for block6.rng      using bits 15 to 24       141650  -.894  .1856
   OPSO for block6.rng      using bits 14 to 23       141720  -.653  .2569
   OPSO for block6.rng      using bits 13 to 22       141570 -1.170  .1210
   OPSO for block6.rng      using bits 12 to 21       141997   .302  .6188
   OPSO for block6.rng      using bits 11 to 20       142171   .902  .8166
   OPSO for block6.rng      using bits 10 to 19       142010   .347  .6358
   OPSO for block6.rng      using bits  9 to 18       141785  -.429  .3341
   OPSO for block6.rng      using bits  8 to 17       141940   .106  .5421
   OPSO for block6.rng      using bits  7 to 16       141881  -.098  .4611
   OPSO for block6.rng      using bits  6 to 15       141823  -.298  .3830
   OPSO for block6.rng      using bits  5 to 14       141844  -.225  .4109
   OPSO for block6.rng      using bits  4 to 13       142206  1.023  .8468
   OPSO for block6.rng      using bits  3 to 12       141934   .085  .5339
   OPSO for block6.rng      using bits  2 to 11       142040   .451  .6739
   OPSO for block6.rng      using bits  1 to 10       142049   .482  .6850
OQSO test for generator block6.rng
 Output: No. missing words (mw), equiv normal variate (z), p-value (p)
                                                         mw     z     p
   OQSO for block6.rng      using bits 28 to 32       141314 -2.018  .0218
   OQSO for block6.rng      using bits 27 to 31       141935   .087  .5347
   OQSO for block6.rng      using bits 26 to 30       142064   .524  .7000
   OQSO for block6.rng      using bits 25 to 29       142213  1.029  .8484
   OQSO for block6.rng      using bits 24 to 28       141289 -2.103  .0177
   OQSO for block6.rng      using bits 23 to 27       141555 -1.201  .1149
   OQSO for block6.rng      using bits 22 to 26       141848  -.208  .4177
   OQSO for block6.rng      using bits 21 to 25       141873  -.123  .4510
   OQSO for block6.rng      using bits 20 to 24       141990   .273  .6078
   OQSO for block6.rng      using bits 19 to 23       142006   .328  .6284
   OQSO for block6.rng      using bits 18 to 22       141602 -1.042  .1488
   OQSO for block6.rng      using bits 17 to 21       142553  2.182  .9854
   OQSO for block6.rng      using bits 16 to 20       142544  2.151  .9843
   OQSO for block6.rng      using bits 15 to 19       141873  -.123  .4510
   OQSO for block6.rng      using bits 14 to 18       141536 -1.266  .1028
   OQSO for block6.rng      using bits 13 to 17       142023   .385  .6500
   OQSO for block6.rng      using bits 12 to 16       141853  -.191  .4243
   OQSO for block6.rng      using bits 11 to 15       142224  1.067  .8569
   OQSO for block6.rng      using bits 10 to 14       142137   .772  .7799
   OQSO for block6.rng      using bits  9 to 13       141535 -1.269  .1022
   OQSO for block6.rng      using bits  8 to 12       142452  1.840  .9671
   OQSO for block6.rng      using bits  7 to 11       141546 -1.232  .1090
   OQSO for block6.rng      using bits  6 to 10       141933   .080  .5320
   OQSO for block6.rng      using bits  5 to  9       141598 -1.055  .1456
   OQSO for block6.rng      using bits  4 to  8       141933   .080  .5320
   OQSO for block6.rng      using bits  3 to  7       141678  -.784  .2165
   OQSO for block6.rng      using bits  2 to  6       141525 -1.303  .0963
   OQSO for block6.rng      using bits  1 to  5       142615  2.392  .9916
 DNA test for generator block6.rng
 Output: No. missing words (mw), equiv normal variate (z), p-value (p)
                                                         mw     z     p
    DNA for block6.rng      using bits 31 to 32       141417 -1.452  .0732
    DNA for block6.rng      using bits 30 to 31       142214   .899  .8156
```

```
DNA for block6.rng      using bits 29 to 30       142038   .380  .6479
DNA for block6.rng      using bits 28 to 29       142208   .881  .8109
DNA for block6.rng      using bits 27 to 28       142426  1.524  .9363
DNA for block6.rng      using bits 26 to 27       141704  -.606  .2724
DNA for block6.rng      using bits 25 to 26       142381  1.391  .9179
DNA for block6.rng      using bits 24 to 25       141701  -.615  .2694
DNA for block6.rng      using bits 23 to 24       142267  1.055  .8543
DNA for block6.rng      using bits 22 to 23       142111   .595  .7240
DNA for block6.rng      using bits 21 to 22       141640  -.794  .2135
DNA for block6.rng      using bits 20 to 21       142003   .276  .6088
DNA for block6.rng      using bits 19 to 20       142226   .934  .8249
DNA for block6.rng      using bits 18 to 19       142126   .639  .7386
DNA for block6.rng      using bits 17 to 18       142097   .554  .7101
DNA for block6.rng      using bits 16 to 17       141257 -1.924  .0272
DNA for block6.rng      using bits 15 to 16       142182   .804  .7894
DNA for block6.rng      using bits 14 to 15       141529 -1.122  .1309
DNA for block6.rng      using bits 13 to 14       141295 -1.812  .0350
DNA for block6.rng      using bits 12 to 13       141449 -1.358  .0872
DNA for block6.rng      using bits 11 to 12       142583  1.987  .9766
DNA for block6.rng      using bits 10 to 11       142362  1.335  .9091
DNA for block6.rng      using bits  9 to 10       142347  1.291  .9017
DNA for block6.rng      using bits  8 to  9       141719  -.561  .2872
DNA for block6.rng      using bits  7 to  8       142138   .675  .7500
DNA for block6.rng      using bits  6 to  7       141281 -1.853  .0319
DNA for block6.rng      using bits  5 to  6       142163   .748  .7729
DNA for block6.rng      using bits  4 to  5       141434 -1.402  .0804
DNA for block6.rng      using bits  3 to  4       141758  -.446  .3277
DNA for block6.rng      using bits  2 to  3       141940   .090  .5360
DNA for block6.rng      using bits  1 to  2       142016   .315  .6235

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
     ::     This is the COUNT-THE-1's TEST on a stream of bytes.      ::
     :: Consider the file under test as a stream of bytes (four per   ::
     :: 32 bit integer).  Each byte can contain from 0 to 8 1's,      ::
     :: with probabilities 1,8,28,56,70,56,28,8,1 over 256.  Now let  ::
     :: the stream of bytes provide a string of overlapping  5-letter ::
     :: words, each "letter" taking values A,B,C,D,E. The letters are ::
     :: determined by the number of 1's in a byte::  0,1,or 2 yield A,::
     :: 3 yields B, 4 yields C, 5 yields D and 6,7 or 8 yield E. Thus ::
     :: we have a monkey at a typewriter hitting five keys with vari- ::
     :: ous probabilities (37,56,70,56,37 over 256).  There are 5^5   ::
     :: possible 5-letter words, and from a string of 256,000 (over-  ::
     :: lapping) 5-letter words, counts are made on the frequencies   ::
     :: for each word.   The quadratic form in the weak inverse of    ::
     :: the covariance matrix of the cell counts provides a chisquare ::
     :: test::  Q5-Q4, the difference of the naive Pearson sums of    ::
     :: (OBS-EXP)^2/EXP on counts for 5- and 4-letter cell counts.    ::
     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
   Test results for block6.rng
 Chi-square with 5^5-5^4=2500 d.of f. for sample size:2560000
                             chisquare  equiv normal  p-value
  Results fo COUNT-THE-1's in successive bytes:
 byte stream for block6.rng       2538.34       .542      .706167
 byte stream for block6.rng       2490.07      -.140      .444178
```

```
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

    :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
    ::     This is the COUNT-THE-1's TEST for specific bytes.        ::
    :: Consider the file under test as a stream of 32-bit integers.  ::
    :: From each integer, a specific byte is chosen , say the left-  ::
    :: most::  bits 1 to 8. Each byte can contain from 0 to 8 1's,   ::
    :: with probabilitie 1,8,28,56,70,56,28,8,1 over 256.  Now let   ::
    :: the specified bytes from successive integers provide a string ::
    :: of (overlapping) 5-letter words, each "letter" taking values  ::
    :: A,B,C,D,E. The letters are determined  by the number of 1's,  ::
    :: in that byte::  0,1,or 2 ---> A, 3 ---> B, 4 ---> C, 5 ---> D,::
    :: and  6,7 or 8 ---> E.  Thus we have a monkey at a typewriter  ::
    :: hitting five keys with with various probabilities::  37,56,70,::
    :: 56,37 over 256. There are 5^5 possible 5-letter words, and    ::
    :: from a string of 256,000 (overlapping) 5-letter words, counts ::
    :: are made on the frequencies for each word. The quadratic form ::
    :: in the weak inverse of the covariance matrix of the cell      ::
    :: counts provides a chisquare test::  Q5-Q4, the difference of  ::
    :: the naive Pearson  sums of (OBS-EXP)^2/EXP on counts for 5-   ::
    :: and 4-letter cell counts.                                     ::
    :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
 Chi-square with 5^5-5^4=2500 d.of f. for sample size: 256000
                     chisquare  equiv normal  p value
  Results for COUNT-THE-1's in specified bytes:
           bits  1 to  8  2419.56     -1.138      .127633
           bits  2 to  9  2512.94       .183      .572592
           bits  3 to 10  2502.42       .034      .513622
           bits  4 to 11  2358.81     -1.997      .022928
           bits  5 to 12  2552.97       .749      .773104
           bits  6 to 13  2418.89     -1.147      .125672
           bits  7 to 14  2541.83       .592      .722935
           bits  8 to 15  2421.25     -1.114      .132696
           bits  9 to 16  2504.60       .065      .525929
           bits 10 to 17  2361.67     -1.956      .025216
           bits 11 to 18  2636.87      1.936      .973543
           bits 12 to 19  2554.43       .770      .779274
           bits 13 to 20  2490.80      -.130      .448249
           bits 14 to 21  2397.33     -1.452      .073247
           bits 15 to 22  2590.43      1.279      .899540
           bits 16 to 23  2452.54      -.671      .251056
           bits 17 to 24  2563.16       .893      .814139
           bits 18 to 25  2469.13      -.437      .331214
           bits 19 to 26  2568.96       .975      .835283
           bits 20 to 27  2468.41      -.447      .327504
           bits 21 to 28  2480.98      -.269      .393993
           bits 22 to 29  2613.70      1.608      .946079
           bits 23 to 30  2635.35      1.914      .972201
           bits 24 to 31  2480.06      -.282      .388962
           bits 25 to 32  2574.81      1.058      .854972

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

    :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
    ::              THIS IS A PARKING LOT TEST                       ::
    :: In a square of side 100, randomly "park" a car---a circle of  ::
    :: radius 1.   Then try to park a 2nd, a 3rd, and so on, each    ::
```

```
:: time parking "by ear".  That is, if an attempt to park a car  ::
:: causes a crash with one already parked, try again at a new    ::
:: random location. (To avoid path problems, consider parking    ::
:: helicopters rather than cars.)   Each attempt leads to either ::
:: a crash or a success, the latter followed by an increment to  ::
:: the list of cars already parked. If we plot n:  the number of ::
:: attempts, versus k::  the number successfully parked, we get a::
:: curve that should be similar to those provided by a perfect   ::
:: random number generator.  Theory for the behavior of such a   ::
:: random curve seems beyond reach, and as graphics displays are ::
:: not available for this battery of tests, a simple characteriz ::
:: ation of the random experiment is used: k, the number of cars ::
:: successfully parked after n=12,000 attempts. Simulation shows ::
:: that k should average 3523 with sigma 21.9 and is very close  ::
:: to normally distributed.  Thus (k-3523)/21.9 should be a st-  ::
:: andard normal variable, which, converted to a uniform varia-  ::
:: ble, provides input to a KSTEST based on a sample of 10.      ::
::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
      CDPARK: result of ten tests on file block6.rng
       Of 12,000 tries, the average no. of successes
            should be 3523 with sigma=21.9
       Successes: 3520    z-score:  -.137 p-value: .445521
       Successes: 3543    z-score:   .913 p-value: .819442
       Successes: 3509    z-score:  -.639 p-value: .261324
       Successes: 3522    z-score:  -.046 p-value: .481790
       Successes: 3501    z-score: -1.005 p-value: .157553
       Successes: 3517    z-score:  -.274 p-value: .392053
       Successes: 3493    z-score: -1.370 p-value: .085365
       Successes: 3527    z-score:   .183 p-value: .572463
       Successes: 3515    z-score:  -.365 p-value: .357445
       Successes: 3491    z-score: -1.461 p-value: .071982

      square size   avg. no.  parked   sample sigma
        100.            3513.800       15.098
       KSTEST for the above 10: p=  .780431

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
::               THE MINIMUM DISTANCE TEST                       ::
:: It does this 100 times::   choose n=8000 random points in a   ::
:: square of side 10000.  Find d, the minimum distance between   ::
:: the (n^2-n)/2 pairs of points.  If the points are truly inde- ::
:: pendent uniform, then d^2, the square of the minimum distance ::
:: should be (very close to) exponentially distributed with mean ::
:: .995 .  Thus 1-exp(-d^2/.995) should be uniform on [0,1) and  ::
:: a KSTEST on the resulting 100 values serves as a test of uni- ::
:: formity for random points in the square. Test numbers=0 mod 5 ::
:: are printed but the KSTEST is based on the full set of 100    ::
:: random choices of 8000 points in the 10000x10000 square.      ::
::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
          This is the MINIMUM DISTANCE test
         for random integers in the file block6.rng
Sample no.    d^2     avg     equiv uni
      5     .8373   1.2477     .568938
     10     .2615    .7447     .231089
     15     .3222    .8470     .276603
```

```
          20     .3293    .7405     .281760
          25     .8094    .7324     .556671
          30    1.5835    .8458     .796370
          35    1.6999    .8546     .818850
          40    1.6143    .8828     .802586
          45     .2280    .8792     .204765
          50     .7023    .9437     .506296
          55     .1686    .8916     .155858
          60     .2770    .9191     .243006
          65     .2577    .9762     .228178
          70     .0445    .9167     .043757
          75     .4939    .9254     .391254
          80    1.2292    .9250     .709286
          85     .9900    .9105     .630264
          90     .1601    .9159     .148615
          95     .0992    .8776     .094906
         100    2.6976    .9206     .933539
     MINIMUM DISTANCE TEST for block6.rng
          Result of KS test on 20 transformed mindist^2's:
                                  p-value= .233572

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

     :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
     ::               THE 3DSPHERES TEST                              ::
     :: Choose  4000 random points in a cube of edge 1000.  At each   ::
     :: point, center a sphere large enough to reach the next closest ::
     :: point. Then the volume of the smallest such sphere is (very   ::
     :: close to) exponentially distributed with mean 120pi/3.  Thus  ::
     :: the radius cubed is exponential with mean 30. (The mean is    ::
     :: obtained by extensive simulation).  The 3DSPHERES test gener- ::
     :: ates 4000 such spheres 20 times.  Each min radius cubed leads ::
     :: to a uniform variable by means of 1-exp(-r^3/30.), then a     ::
     ::  KSTEST is done on the 20 p-values.                           ::
     :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
               The 3DSPHERES test for file block6.rng
 sample no:  1     r^3=  54.287     p-value= .83627
 sample no:  2     r^3=  30.124     p-value= .63364
 sample no:  3     r^3=  28.748     p-value= .61644
 sample no:  4     r^3=   4.407     p-value= .13663
 sample no:  5     r^3=   4.243     p-value= .13189
 sample no:  6     r^3=  26.822     p-value= .59101
 sample no:  7     r^3=  46.478     p-value= .78759
 sample no:  8     r^3=  12.604     p-value= .34303
 sample no:  9     r^3=  34.737     p-value= .68586
 sample no: 10     r^3=   8.896     p-value= .25661
 sample no: 11     r^3=    .308     p-value= .01022
 sample no: 12     r^3=  37.168     p-value= .71031
 sample no: 13     r^3=   6.282     p-value= .18893
 sample no: 14     r^3=  11.910     p-value= .32766
 sample no: 15     r^3=  44.415     p-value= .77248
 sample no: 16     r^3=  34.185     p-value= .68002
 sample no: 17     r^3=  38.573     p-value= .72356
 sample no: 18     r^3=  45.426     p-value= .78002
 sample no: 19     r^3=   8.581     p-value= .24876
 sample no: 20     r^3=   5.272     p-value= .16117
  A KS test is applied to those 20 p-values.
```

```
---------------------------------------------------------
       3DSPHERES test for file block6.rng           p-value= .461227
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

     :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
     ::      This is the SQEEZE test                                  ::
     ::  Random integers are floated to get uniforms on [0,1). Start- ::
     ::  ing with k=2^31=2147483647, the test finds j, the number of  ::
     ::  iterations necessary to reduce k to 1, using the reduction   ::
     ::  k=ceiling(k*U), with U provided by floating integers from    ::
     ::  the file being tested.  Such j's are found 100,000 times,    ::
     ::  then counts for the number of times j was <=6,7,...,47,>=48  ::
     ::  are used to provide a chi-square test for cell frequencies.  ::
     :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
            RESULTS OF SQUEEZE TEST FOR block6.rng
         Table of standardized frequency counts
     ( (obs-exp)/sqrt(exp) )^2
        for j taking values <=6,7,8,...,47,>=48:
    -1.5     -.7     -.1    -1.6     2.6      .3
      .5      .8      .0     1.1     -.8     -.8
      .9      .7    -1.6     1.4      .2     -.3
     -.5    -1.1     -.6      .4     -.3     2.8
    -1.5     -.2     -.1     -.4     -.9      .0
      .5     2.7    -1.1     -.3     -.1      .8
      .5    -1.3      .1      .4      .9     2.0
     -.1
           Chi-square with 42 degrees of freedom: 50.856
              z-score=   .966  p-value= .835971
______________________________________________________________

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

     :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
     ::             The  OVERLAPPING SUMS test                        ::
     :: Integers are floated to get a sequence U(1),U(2),... of uni-  ::
     :: form [0,1) variables.  Then overlapping sums,                 ::
     ::   S(1)=U(1)+...+U(100), S2=U(2)+...+U(101),... are formed.    ::
     :: The S's are virtually normal with a certain covariance mat-   ::
     :: rix.  A linear transformation of the S's converts them to a   ::
     :: sequence of independent standard normals, which are converted ::
     :: to uniform variables for a KSTEST. The  p-values from ten     ::
     :: KSTESTs are given still another KSTEST.                       ::
     :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
                Test no.  1      p-value  .467835
                Test no.  2      p-value  .684353
                Test no.  3      p-value  .765742
                Test no.  4      p-value  .667671
                Test no.  5      p-value  .748993
                Test no.  6      p-value  .676145
                Test no.  7      p-value  .356360
                Test no.  8      p-value  .606335
                Test no.  9      p-value  .297356
                Test no. 10      p-value  .757876
   Results of the OSUM test for block6.rng
        KSTEST on the above 10 p-values:  .870741

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
```

```
::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
::     This is the RUNS test.  It counts runs up, and runs down, ::
:: in a sequence of uniform [0,1) variables, obtained by float-  ::
:: ing the 32-bit integers in the specified file. This example   ::
:: shows how runs are counted:  .123,.357,.789,.425,.224,.416,.95::
:: contains an up-run of length 3, a down-run of length 2 and an ::
:: up-run of (at least) 2, depending on the next values.  The    ::
:: covariance matrices for the runs-up and runs-down are well    ::
:: known, leading to chisquare tests for quadratic forms in the  ::
:: weak inverses of the covariance matrices.  Runs are counted   ::
:: for sequences of length 10,000.  This is done ten times. Then ::
:: repeated.                                                     ::
::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
      The RUNS test for file block6.rng
Up and down runs in a sample of 10000
```

```
                 Run test for block6.rng     :
      runs up; ks test for 10 p's: .095465
    runs down; ks test for 10 p's: .728443
                 Run test for block6.rng     :
      runs up; ks test for 10 p's: .408193
    runs down; ks test for 10 p's: .164630

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
:: This is the CRAPS TEST. It plays 200,000 games of craps, finds::
:: the number of wins and the number of throws necessary to end  ::
:: each game.  The number of wins should be (very close to) a    ::
:: normal with mean 200000p and variance 200000p(1-p), with      ::
:: p=244/495.  Throws necessary to complete the game can vary    ::
:: from 1 to infinity, but counts for all>21 are lumped with 21. ::
:: A chi-square test is made on the no.-of-throws cell counts.   ::
:: Each 32-bit integer from the test file provides the value for ::
:: the throw of a die, by floating to [0,1), multiplying by 6    ::
:: and taking 1 plus the integer part of the result.             ::
::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
                Results of craps test for block6.rng
  No. of wins:  Observed Expected
                                98798    98585.86
                  98798= No. of wins, z-score=  .949 pvalue= .82864
   Analysis of Throws-per-Game:
 Chisq=  20.19 for 20 degrees of freedom, p=  .55382
               Throws Observed Expected  Chisq     Sum
                  1    66478    66666.7    .534     .534
                  2    37973    37654.3   2.697    3.231
                  3    26936    26954.7    .013    3.244
                  4    19347    19313.5    .058    3.302
                  5    13745    13851.4    .818    4.120
                  6     9827     9943.5   1.366    5.486
                  7     7157     7145.0    .020    5.506
                  8     5136     5139.1    .002    5.508
                  9     3695     3699.9    .006    5.514
                 10     2603     2666.3   1.503    7.017
                 11     1921     1923.3    .003    7.020
                 12     1456     1388.7   3.258   10.277
```

```
               13     1036     1003.7   1.038   11.316
               14      719      726.1    .070   11.386
               15      571      525.8   3.879   15.265
               16      376      381.2    .070   15.335
               17      257      276.5   1.381   16.715
               18      189      200.8    .697   17.412
               19      156      146.0    .687   18.099
               20      114      106.2    .571   18.670
               21      308      287.1   1.519   20.189
          SUMMARY  FOR block6.rng
              p-value for no. of wins: .828643
              p-value for throws/game: .553823

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

 Results of DIEHARD battery of tests sent to file report6.txt



       NOTE: Most of the tests in DIEHARD return a p-value, which
       should be uniform on [0,1) if the input file contains truly
       independent random bits.   Those p-values are obtained by
       p=F(X), where F is the assumed distribution of the sample
       random variable X---often normal. But that assumed F is just
       an asymptotic approximation, for which the fit will be worst
       in the tails. Thus you should not be surprised with
       occasional p-values near 0 or 1, such as .0012 or .9983.
       When a bit stream really FAILS BIG, you will get p's of 0 or
       1 to six or more places.  By all means, do not, as a
       Statistician might, think that a p < .025 or p> .975 means
       that the RNG has "failed the test at the .05 level".  Such
       p's happen among the hundreds that DIEHARD produces, even
       with good RNG's.  So keep in mind that " p happens".
     :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
     ::            This is the BIRTHDAY SPACINGS TEST                 ::
     :: Choose m birthdays in a year of n days.  List the spacings    ::
     :: between the birthdays.  If j is the number of values that     ::
     :: occur more than once in that list, then j is asymptotically   ::
     :: Poisson distributed with mean m^3/(4n).  Experience shows n   ::
     :: must be quite large, say n>=2^18, for comparing the results   ::
     :: to the Poisson distribution with that mean.  This test uses   ::
     :: n=2^24 and m=2^9,  so that the underlying distribution for j  ::
     :: is taken to be Poisson with lambda=2^27/(2^26)=2.  A sample   ::
     :: of 500 j's is taken, and a chi-square goodness of fit test    ::
     :: provides a p value.  The first test uses bits 1-24 (counting  ::
     :: from the left) from integers in the specified file.           ::
     ::   Then the file is closed and reopened. Next, bits 2-25 are   ::
     :: used to provide birthdays, then 3-26 and so on to bits 9-32.  ::
     :: Each set of bits provides a p-value, and the nine p-values    ::
     :: provide a sample for a KSTEST.                                ::
     :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
 BIRTHDAY SPACINGS TEST, M= 512 N=2**24 LAMBDA=  2.0000
           Results for block7.rng
                   For a sample of size 500:     mean
           block7.rng      using bits  1 to 24   2.020
  duplicate       number       number
```

```
 spacings       observed     expected
      0           70.       67.668
      1          131.      135.335
      2          139.      135.335
      3           88.       90.224
      4           40.       45.112
      5           21.       18.045
 6 to INF         11.        8.282
Chisquare with  6 d.o.f. =     2.33 p-value=  .112873
 :::::::::::::::::::::::::::::::::::::::::
                  For a sample of size 500:     mean
          block7.rng      using bits  2 to 25   1.958
 duplicate       number       number
 spacings       observed     expected
      0           69.       67.668
      1          135.      135.335
      2          137.      135.335
      3           89.       90.224
      4           50.       45.112
      5           17.       18.045
 6 to INF          3.        8.282
Chisquare with  6 d.o.f. =     4.02 p-value=  .326413
 :::::::::::::::::::::::::::::::::::::::::
                  For a sample of size 500:     mean
          block7.rng      using bits  3 to 26   2.060
 duplicate       number       number
 spacings       observed     expected
      0           67.       67.668
      1          129.      135.335
      2          143.      135.335
      3           81.       90.224
      4           47.       45.112
      5           18.       18.045
 6 to INF         15.        8.282
Chisquare with  6 d.o.f. =     7.21 p-value=  .698057
 :::::::::::::::::::::::::::::::::::::::::
                  For a sample of size 500:     mean
          block7.rng      using bits  4 to 27   1.970
 duplicate       number       number
 spacings       observed     expected
      0           79.       67.668
      1          127.      135.335
      2          126.      135.335
      3           96.       90.224
      4           51.       45.112
      5           13.       18.045
 6 to INF          8.        8.282
Chisquare with  6 d.o.f. =     5.61 p-value=  .532150
 :::::::::::::::::::::::::::::::::::::::::
                  For a sample of size 500:     mean
          block7.rng      using bits  5 to 28   2.050
 duplicate       number       number
 spacings       observed     expected
      0           58.       67.668
      1          141.      135.335
      2          139.      135.335
      3           85.       90.224
```

```
      4          46.       45.112
      5          21.       18.045
 6 to INF        10.        8.282
Chisquare with  6 d.o.f. =     2.88 p-value=  .175996
 :::::::::::::::::::::::::::::::::::::::::
                 For a sample of size 500:     mean
         block7.rng      using bits  6 to 29   2.014
 duplicate       number       number
 spacings       observed     expected
      0          65.       67.668
      1         134.      135.335
      2         132.      135.335
      3          87.       90.224
      4          69.       45.112
      5           7.       18.045
 6 to INF         6.        8.282
Chisquare with  6 d.o.f. =    20.35 p-value=  .997605
 :::::::::::::::::::::::::::::::::::::::::
                 For a sample of size 500:     mean
         block7.rng      using bits  7 to 30   1.938
 duplicate       number       number
 spacings       observed     expected
      0          72.       67.668
      1         132.      135.335
      2         142.      135.335
      3          92.       90.224
      4          41.       45.112
      5          15.       18.045
 6 to INF         6.        8.282
Chisquare with  6 d.o.f. =     2.24 p-value=  .103639
 :::::::::::::::::::::::::::::::::::::::::
                 For a sample of size 500:     mean
         block7.rng      using bits  8 to 31   1.948
 duplicate       number       number
 spacings       observed     expected
      0          57.       67.668
      1         150.      135.335
      2         142.      135.335
      3          97.       90.224
      4          30.       45.112
      5          19.       18.045
 6 to INF         5.        8.282
Chisquare with  6 d.o.f. =    10.52 p-value=  .895650
 :::::::::::::::::::::::::::::::::::::::::
                 For a sample of size 500:     mean
         block7.rng      using bits  9 to 32   1.978
 duplicate       number       number
 spacings       observed     expected
      0          65.       67.668
      1         142.      135.335
      2         138.      135.335
      3          83.       90.224
      4          44.       45.112
      5          24.       18.045
 6 to INF         4.        8.282
Chisquare with  6 d.o.f. =     5.27 p-value=  .490415
 :::::::::::::::::::::::::::::::::::::::::
```

```
   The 9 p-values were
        .112873   .326413   .698057   .532150   .175996
        .997605   .103639   .895650   .490415
  A KSTEST for the 9 p-values yields  .282488

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

     :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
     ::            THE OVERLAPPING 5-PERMUTATION TEST                 ::
     :: This is the OPERM5 test.  It looks at a sequence of one mill- ::
     :: ion 32-bit random integers.  Each set of five consecutive     ::
     :: integers can be in one of 120 states, for the 5! possible or- ::
     :: derings of five numbers.  Thus the 5th, 6th, 7th,...numbers   ::
     :: each provide a state. As many thousands of state transitions  ::
     :: are observed,  cumulative counts are made of the number of    ::
     :: occurences of each state.  Then the quadratic form in the     ::
     :: weak inverse of the 120x120 covariance matrix yields a test   ::
     :: equivalent to the likelihood ratio test that the 120 cell     ::
     :: counts came from the specified (asymptotically) normal dis-   ::
     :: tribution with the specified 120x120 covariance matrix (with  ::
     :: rank 99).  This version uses 1,000,000 integers, twice.       ::
     :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
           OPERM5 test for file block7.rng
     For a sample of 1,000,000 consecutive 5-tuples,
 chisquare for 99 degrees of freedom= 79.893; p-value= .079479
           OPERM5 test for file block7.rng
     For a sample of 1,000,000 consecutive 5-tuples,
 chisquare for 99 degrees of freedom=114.672; p-value= .865868
     :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
     :: This is the BINARY RANK TEST for 31x31 matrices. The leftmost ::
     :: 31 bits of 31 random integers from the test sequence are used ::
     :: to form a 31x31 binary matrix over the field {0,1}. The rank  ::
     :: is determined. That rank can be from 0 to 31, but ranks< 28   ::
     :: are rare, and their counts are pooled with those for rank 28. ::
     :: Ranks are found for 40,000 such random matrices and a chisqua-::
     :: re test is performed on counts for ranks 31,30,29 and <=28.   ::
     :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
    Binary rank test for block7.rng
         Rank test for 31x31 binary matrices:
        rows from leftmost 31 bits of each 32-bit integer
      rank   observed  expected (o-e)^2/e  sum
        28       222     211.4   .529654     .530
        29      5258    5134.0  2.994434    3.524
        30     22990   23103.0   .553156    4.077
        31     11530   11551.5   .040107    4.117
  chisquare= 4.117 for 3 d. of f.; p-value= .772155
--------------------------------------------------------------
     :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
     :: This is the BINARY RANK TEST for 32x32 matrices. A random 32x ::
     :: 32 binary matrix is formed, each row a 32-bit random integer. ::
     :: The rank is determined. That rank can be from 0 to 32, ranks  ::
     :: less than 29 are rare, and their counts are pooled with those ::
     :: for rank 29.  Ranks are found for 40,000 such random matrices ::
     :: and a chisquare test is performed on counts for ranks  32,31, ::
     :: 30 and <=29.                                                  ::
     :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
    Binary rank test for block7.rng
```

```
         Rank test for 32x32 binary matrices:
        rows from leftmost 32 bits of each 32-bit integer
      rank   observed  expected (o-e)^2/e  sum
        29       231     211.4  1.813725    1.814
        30      5173    5134.0   .296104    2.110
        31     22996   23103.0   .495997    2.606
        32     11600   11551.5   .203426    2.809
  chisquare= 2.809 for 3 d. of f.; p-value= .624882
--------------------------------------------------------------

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
     :: This is the BINARY RANK TEST for 6x8 matrices.  From each of  ::
     :: six random 32-bit integers from the generator under test, a   ::
     :: specified byte is chosen, and the resulting six bytes form a  ::
     :: 6x8 binary matrix whose rank is determined.  That rank can be ::
     :: from 0 to 6, but ranks 0,1,2,3 are rare; their counts are     ::
     :: pooled with those for rank 4. Ranks are found for 100,000     ::
     :: random matrices, and a chi-square test is performed on        ::
     :: counts for ranks 6,5 and <=4.                                 ::
     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
         Binary Rank Test for block7.rng
        Rank of a 6x8 binary matrix,
     rows formed from eight bits of the RNG block7.rng
     b-rank test for bits  1 to  8
                     OBSERVED   EXPECTED     (O-E)^2/E      SUM
          r<=4          943       944.3         .002         .002
          r =5        21369     21743.9        6.464        6.466
          r =6        77688     77311.8        1.831        8.296
                        p=1-exp(-SUM/2)= .98421
        Rank of a 6x8 binary matrix,
     rows formed from eight bits of the RNG block7.rng
     b-rank test for bits  2 to  9
                     OBSERVED   EXPECTED     (O-E)^2/E      SUM
          r<=4          947       944.3         .008         .008
          r =5        21792     21743.9         .106         .114
          r =6        77261     77311.8         .033         .147
                        p=1-exp(-SUM/2)= .07110
        Rank of a 6x8 binary matrix,
     rows formed from eight bits of the RNG block7.rng
     b-rank test for bits  3 to 10
                     OBSERVED   EXPECTED     (O-E)^2/E      SUM
          r<=4         1047       944.3       11.169       11.169
          r =5        21753     21743.9         .004       11.173
          r =6        77200     77311.8         .162       11.335
                        p=1-exp(-SUM/2)= .99654
        Rank of a 6x8 binary matrix,
     rows formed from eight bits of the RNG block7.rng
     b-rank test for bits  4 to 11
                     OBSERVED   EXPECTED     (O-E)^2/E      SUM
          r<=4          925       944.3         .395         .395
          r =5        21690     21743.9         .134         .528
          r =6        77385     77311.8         .069         .597
                        p=1-exp(-SUM/2)= .25823
        Rank of a 6x8 binary matrix,
     rows formed from eight bits of the RNG block7.rng
```

```
b-rank test for bits  5 to 12
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          879       944.3       4.516       4.516
     r =5        21669     21743.9        .258       4.774
     r =6        77452     77311.8        .254       5.028
                   p=1-exp(-SUM/2)= .91906
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block7.rng
b-rank test for bits  6 to 13
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          918       944.3        .733        .733
     r =5        21808     21743.9        .189        .922
     r =6        77274     77311.8        .018        .940
                   p=1-exp(-SUM/2)= .37500
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block7.rng
b-rank test for bits  7 to 14
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          883       944.3       3.979       3.979
     r =5        21599     21743.9        .966       4.945
     r =6        77518     77311.8        .550       5.495
                   p=1-exp(-SUM/2)= .93591
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block7.rng
b-rank test for bits  8 to 15
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          928       944.3        .281        .281
     r =5        21857     21743.9        .588        .870
     r =6        77215     77311.8        .121        .991
                   p=1-exp(-SUM/2)= .39070
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block7.rng
b-rank test for bits  9 to 16
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4         1004       944.3       3.774       3.774
     r =5        21616     21743.9        .752       4.526
     r =6        77380     77311.8        .060       4.587
                   p=1-exp(-SUM/2)= .89907
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block7.rng
b-rank test for bits 10 to 17
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          943       944.3        .002        .002
     r =5        21896     21743.9       1.064       1.066
     r =6        77161     77311.8        .294       1.360
                   p=1-exp(-SUM/2)= .49336
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block7.rng
b-rank test for bits 11 to 18
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          907       944.3       1.473       1.473
     r =5        21836     21743.9        .390       1.864
     r =6        77257     77311.8        .039       1.902
                   p=1-exp(-SUM/2)= .61372
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block7.rng
b-rank test for bits 12 to 19
```

```
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          941        944.3        .012         .012
     r =5        21578      21743.9       1.266        1.277
     r =6        77481      77311.8        .370        1.648
                   p=1-exp(-SUM/2)= .56124
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block7.rng
b-rank test for bits 13 to 20
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          958        944.3        .199         .199
     r =5        21770      21743.9        .031         .230
     r =6        77272      77311.8        .020         .251
                   p=1-exp(-SUM/2)= .11774
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block7.rng
b-rank test for bits 14 to 21
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          958        944.3        .199         .199
     r =5        21760      21743.9        .012         .211
     r =6        77282      77311.8        .011         .222
                   p=1-exp(-SUM/2)= .10512
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block7.rng
b-rank test for bits 15 to 22
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          994        944.3       2.616        2.616
     r =5        21878      21743.9        .827        3.443
     r =6        77128      77311.8        .437        3.880
                   p=1-exp(-SUM/2)= .85627
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block7.rng
b-rank test for bits 16 to 23
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          932        944.3        .160         .160
     r =5        21798      21743.9        .135         .295
     r =6        77270      77311.8        .023         .317
                   p=1-exp(-SUM/2)= .14677
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block7.rng
b-rank test for bits 17 to 24
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          984        944.3       1.669        1.669
     r =5        21527      21743.9       2.164        3.833
     r =6        77489      77311.8        .406        4.239
                   p=1-exp(-SUM/2)= .87989
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block7.rng
b-rank test for bits 18 to 25
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          983        944.3       1.586        1.586
     r =5        21876      21743.9        .803        2.388
     r =6        77141      77311.8        .377        2.766
                   p=1-exp(-SUM/2)= .74915
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG block7.rng
b-rank test for bits 19 to 26
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
```

```
       r<=4          938       944.3         .042         .042
       r =5        21833     21743.9         .365         .407
       r =6        77229     77311.8         .089         .496
                     p=1-exp(-SUM/2)= .21958
     Rank of a 6x8 binary matrix,
  rows formed from eight bits of the RNG block7.rng
  b-rank test for bits 20 to 27
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
       r<=4          949       944.3         .023         .023
       r =5        21765     21743.9         .020         .044
       r =6        77286     77311.8         .009         .052
                     p=1-exp(-SUM/2)= .02589
     Rank of a 6x8 binary matrix,
  rows formed from eight bits of the RNG block7.rng
  b-rank test for bits 21 to 28
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
       r<=4          999       944.3        3.168        3.168
       r =5        21819     21743.9         .259        3.428
       r =6        77182     77311.8         .218        3.646
                     p=1-exp(-SUM/2)= .83844
     Rank of a 6x8 binary matrix,
  rows formed from eight bits of the RNG block7.rng
  b-rank test for bits 22 to 29
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
       r<=4          950       944.3         .034         .034
       r =5        21826     21743.9         .310         .344
       r =6        77224     77311.8         .100         .444
                     p=1-exp(-SUM/2)= .19912
     Rank of a 6x8 binary matrix,
  rows formed from eight bits of the RNG block7.rng
  b-rank test for bits 23 to 30
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
       r<=4          963       944.3         .370         .370
       r =5        21713     21743.9         .044         .414
       r =6        77324     77311.8         .002         .416
                     p=1-exp(-SUM/2)= .18783
     Rank of a 6x8 binary matrix,
  rows formed from eight bits of the RNG block7.rng
  b-rank test for bits 24 to 31
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
       r<=4          988       944.3        2.022        2.022
       r =5        21864     21743.9         .663        2.686
       r =6        77148     77311.8         .347        3.033
                     p=1-exp(-SUM/2)= .78048
     Rank of a 6x8 binary matrix,
  rows formed from eight bits of the RNG block7.rng
  b-rank test for bits 25 to 32
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
       r<=4          946       944.3         .003         .003
       r =5        21617     21743.9         .741         .744
       r =6        77437     77311.8         .203         .946
                     p=1-exp(-SUM/2)= .37699
   TEST SUMMARY, 25 tests on 100,000 random 6x8 matrices
 These should be 25 uniform [0,1] random variables:
    .984206     .071096     .996543     .258227     .919057
    .375000     .935914     .390703     .899069     .493357
    .613724     .561239     .117744     .105121     .856272
```

```
     .146771     .879891     .749153     .219577     .025893
     .838439     .199125     .187835     .780481     .376994
   brank test summary for block7.rng
       The KS test for those 25 supposed UNI's yields
                    KS p-value= .558088

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

     :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
     ::                   THE BITSTREAM TEST                          ::
     :: The file under test is viewed as a stream of bits. Call them  ::
     :: b1,b2,... .  Consider an alphabet with two "letters", 0 and 1 ::
     :: and think of the stream of bits as a succession of 20-letter  ::
     :: "words", overlapping.  Thus the first word is b1b2...b20, the ::
     :: second is b2b3...b21, and so on.  The bitstream test counts   ::
     :: the number of missing 20-letter (20-bit) words in a string of ::
     :: 2^21 overlapping 20-letter words.  There are 2^20 possible 20 ::
     :: letter words.  For a truly random string of 2^21+19 bits, the ::
     :: number of missing words j should be (very close to) normally  ::
     :: distributed with mean 141,909 and sigma 428.  Thus            ::
     ::  (j-141909)/428 should be a standard normal variate (z score) ::
     :: that leads to a uniform [0,1) p value.  The test is repeated  ::
     :: twenty times.                                                 ::
     :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
 THE OVERLAPPING 20-tuples BITSTREAM  TEST, 20 BITS PER WORD, N words
   This test uses N=2^21 and samples the bitstream 20 times.
  No. missing words should average  141909. with sigma=428.
---------------------------------------------------------
 tst no  1:  141621 missing words,    -.67 sigmas from mean, p-value= .25026
 tst no  2:  141950 missing words,     .10 sigmas from mean, p-value= .53785
 tst no  3:  141541 missing words,    -.86 sigmas from mean, p-value= .19473
 tst no  4:  142871 missing words,    2.25 sigmas from mean, p-value= .98768
 tst no  5:  141595 missing words,    -.73 sigmas from mean, p-value= .23135
 tst no  6:  141722 missing words,    -.44 sigmas from mean, p-value= .33081
 tst no  7:  141861 missing words,    -.11 sigmas from mean, p-value= .45505
 tst no  8:  142415 missing words,    1.18 sigmas from mean, p-value= .88129
 tst no  9:  141673 missing words,    -.55 sigmas from mean, p-value= .29042
 tst no 10:  141273 missing words,   -1.49 sigmas from mean, p-value= .06854
 tst no 11:  141683 missing words,    -.53 sigmas from mean, p-value= .29847
 tst no 12:  141061 missing words,   -1.98 sigmas from mean, p-value= .02374
 tst no 13:  142007 missing words,     .23 sigmas from mean, p-value= .59026
 tst no 14:  141330 missing words,   -1.35 sigmas from mean, p-value= .08794
 tst no 15:  142562 missing words,    1.52 sigmas from mean, p-value= .93636
 tst no 16:  142229 missing words,     .75 sigmas from mean, p-value= .77244
 tst no 17:  142178 missing words,     .63 sigmas from mean, p-value= .73491
 tst no 18:  142172 missing words,     .61 sigmas from mean, p-value= .73030
 tst no 19:  141751 missing words,    -.37 sigmas from mean, p-value= .35572
 tst no 20:  142016 missing words,     .25 sigmas from mean, p-value= .59841

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

     :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
     ::             The tests OPSO, OQSO and DNA                      ::
     ::         OPSO means Overlapping-Pairs-Sparse-Occupancy         ::
     :: The OPSO test considers 2-letter words from an alphabet of    ::
     :: 1024 letters.  Each letter is determined by a specified ten   ::
     :: bits from a 32-bit integer in the sequence to be tested. OPSO ::
```

```
:: generates  2^21 (overlapping) 2-letter words  (from 2^21+1    ::
:: "keystrokes")  and counts the number of missing words---that  ::
:: is 2-letter words which do not appear in the entire sequence. ::
:: That count should be very close to normally distributed with  ::
:: mean 141,909, sigma 290. Thus (missingwrds-141909)/290 should ::
:: be a standard normal variable. The OPSO test takes 32 bits at ::
:: a time from the test file and uses a designated set of ten    ::
:: consecutive bits. It then restarts the file for the next de-  ::
:: signated 10 bits, and so on.                                  ::
::                                                               ::
::     OQSO means Overlapping-Quadruples-Sparse-Occupancy        ::
::   The test OQSO is similar, except that it considers 4-letter ::
:: words from an alphabet of 32 letters, each letter determined  ::
:: by a designated string of 5 consecutive bits from the test    ::
:: file, elements of which are assumed 32-bit random integers.   ::
:: The mean number of missing words in a sequence of 2^21 four-  ::
:: letter words,  (2^21+3 "keystrokes"), is again 141909, with   ::
:: sigma = 295.  The mean is based on theory; sigma comes from   ::
:: extensive simulation.                                         ::
::                                                               ::
::    The DNA test considers an alphabet of 4 letters::  C,G,A,T,::
:: determined by two designated bits in the sequence of random   ::
:: integers being tested.  It considers 10-letter words, so that ::
:: as in OPSO and OQSO, there are 2^20 possible words, and the   ::
:: mean number of missing words from a string of 2^21  (over-    ::
:: lapping)  10-letter  words (2^21+9 "keystrokes") is 141909.   ::
:: The standard deviation sigma=339 was determined as for OQSO   ::
:: by simulation.  (Sigma for OPSO, 290, is the true value (to   ::
:: three places), not determined by simulation.                  ::
:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
OPSO test for generator block7.rng
 Output: No. missing words (mw), equiv normal variate (z), p-value (p)
                                                          mw     z     p
   OPSO for block7.rng      using bits 23 to 32       142031   .420  .6626
   OPSO for block7.rng      using bits 22 to 31       141892  -.060  .4762
   OPSO for block7.rng      using bits 21 to 30       142042   .457  .6763
   OPSO for block7.rng      using bits 20 to 29       141920   .037  .5147
   OPSO for block7.rng      using bits 19 to 28       142114   .706  .7598
   OPSO for block7.rng      using bits 18 to 27       142297  1.337  .9094
   OPSO for block7.rng      using bits 17 to 26       142336  1.471  .9294
   OPSO for block7.rng      using bits 16 to 25       141936   .092  .5366
   OPSO for block7.rng      using bits 15 to 24       142349  1.516  .9353
   OPSO for block7.rng      using bits 14 to 23       141599 -1.070  .1423
   OPSO for block7.rng      using bits 13 to 22       141507 -1.387  .0827
   OPSO for block7.rng      using bits 12 to 21       141716  -.667  .2525
   OPSO for block7.rng      using bits 11 to 20       141735  -.601  .2739
   OPSO for block7.rng      using bits 10 to 19       141866  -.149  .4406
   OPSO for block7.rng      using bits  9 to 18       141789  -.415  .3391
   OPSO for block7.rng      using bits  8 to 17       142165   .882  .8110
   OPSO for block7.rng      using bits  7 to 16       142153   .840  .7996
   OPSO for block7.rng      using bits  6 to 15       141699  -.725  .2341
   OPSO for block7.rng      using bits  5 to 14       141682  -.784  .2166
   OPSO for block7.rng      using bits  4 to 13       142174   .913  .8193
   OPSO for block7.rng      using bits  3 to 12       142108   .685  .7534
   OPSO for block7.rng      using bits  2 to 11       141767  -.491  .3118
   OPSO for block7.rng      using bits  1 to 10       142071   .557  .7114
OQSO test for generator block7.rng
```

```
Output: No. missing words (mw), equiv normal variate (z), p-value (p)
                                                        mw      z     p
  OQSO for block7.rng      using bits 28 to 32        141417 -1.669  .0476
  OQSO for block7.rng      using bits 27 to 31        141885  -.082  .4671
  OQSO for block7.rng      using bits 26 to 30        141679  -.781  .2175
  OQSO for block7.rng      using bits 25 to 29        142498  1.995  .9770
  OQSO for block7.rng      using bits 24 to 28        142383  1.606  .9458
  OQSO for block7.rng      using bits 23 to 27        141743  -.564  .2864
  OQSO for block7.rng      using bits 22 to 26        141650  -.879  .1897
  OQSO for block7.rng      using bits 21 to 25        142042   .450  .6735
  OQSO for block7.rng      using bits 20 to 24        142112   .687  .7540
  OQSO for block7.rng      using bits 19 to 23        142479  1.931  .9733
  OQSO for block7.rng      using bits 18 to 22        142322  1.399  .9191
  OQSO for block7.rng      using bits 17 to 21        142163   .860  .8051
  OQSO for block7.rng      using bits 16 to 20        141799  -.374  .3542
  OQSO for block7.rng      using bits 15 to 19        141851  -.198  .4216
  OQSO for block7.rng      using bits 14 to 18        141496 -1.401  .0806
  OQSO for block7.rng      using bits 13 to 17        141855  -.184  .4269
  OQSO for block7.rng      using bits 12 to 16        141909  -.001  .4996
  OQSO for block7.rng      using bits 11 to 15        141766  -.486  .3135
  OQSO for block7.rng      using bits 10 to 14        141374 -1.815  .0348
  OQSO for block7.rng      using bits  9 to 13        141707  -.686  .2464
  OQSO for block7.rng      using bits  8 to 12        141973   .216  .5854
  OQSO for block7.rng      using bits  7 to 11        142142   .789  .7849
  OQSO for block7.rng      using bits  6 to 10        141584 -1.103  .1351
  OQSO for block7.rng      using bits  5 to  9        141951   .141  .5562
  OQSO for block7.rng      using bits  4 to  8        141504 -1.374  .0847
  OQSO for block7.rng      using bits  3 to  7        141547 -1.228  .1097
  OQSO for block7.rng      using bits  2 to  6        141632  -.940  .1736
  OQSO for block7.rng      using bits  1 to  5        141813  -.327  .3720
DNA test for generator block7.rng
Output: No. missing words (mw), equiv normal variate (z), p-value (p)
                                                        mw      z     p
   DNA for block7.rng      using bits 31 to 32        141220 -2.033  .0210
   DNA for block7.rng      using bits 30 to 31        141515 -1.163  .1224
   DNA for block7.rng      using bits 29 to 30        141450 -1.355  .0877
   DNA for block7.rng      using bits 28 to 29        141584  -.960  .1686
   DNA for block7.rng      using bits 27 to 28        141795  -.337  .3680
   DNA for block7.rng      using bits 26 to 27        142217   .908  .8180
   DNA for block7.rng      using bits 25 to 26        141548 -1.066  .1432
   DNA for block7.rng      using bits 24 to 25        141503 -1.199  .1153
   DNA for block7.rng      using bits 23 to 24        142025   .341  .6335
   DNA for block7.rng      using bits 22 to 23        141775  -.396  .3460
   DNA for block7.rng      using bits 21 to 22        141777  -.390  .3481
   DNA for block7.rng      using bits 20 to 21        142388  1.412  .9210
   DNA for block7.rng      using bits 19 to 20        142100   .562  .7131
   DNA for block7.rng      using bits 18 to 19        141539 -1.092  .1373
   DNA for block7.rng      using bits 17 to 18        141819  -.266  .3949
   DNA for block7.rng      using bits 16 to 17        141949   .117  .5466
   DNA for block7.rng      using bits 15 to 16        141814  -.281  .3893
   DNA for block7.rng      using bits 14 to 15        141894  -.045  .4820
   DNA for block7.rng      using bits 13 to 14        141978   .203  .5803
   DNA for block7.rng      using bits 12 to 13        141489 -1.240  .1075
   DNA for block7.rng      using bits 11 to 12        141975   .194  .5768
   DNA for block7.rng      using bits 10 to 11        142183   .807  .7903
   DNA for block7.rng      using bits  9 to 10        141831  -.231  .4086
   DNA for block7.rng      using bits  8 to  9        141597  -.921  .1784
```

```
    DNA for block7.rng      using bits  7 to  8        141514 -1.166  .1218
    DNA for block7.rng      using bits  6 to  7        141890  -.057  .4773
    DNA for block7.rng      using bits  5 to  6        141752  -.464  .3213
    DNA for block7.rng      using bits  4 to  5        142441  1.568  .9416
    DNA for block7.rng      using bits  3 to  4        141630  -.824  .2050
    DNA for block7.rng      using bits  2 to  3        142319  1.208  .8866
    DNA for block7.rng      using bits  1 to  2        141620  -.853  .1967

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

    ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
    ::     This is the COUNT-THE-1's TEST on a stream of bytes.       ::
    :: Consider the file under test as a stream of bytes (four per    ::
    :: 32 bit integer).  Each byte can contain from 0 to 8 1's,       ::
    :: with probabilities 1,8,28,56,70,56,28,8,1 over 256.  Now let   ::
    :: the stream of bytes provide a string of overlapping  5-letter  ::
    :: words, each "letter" taking values A,B,C,D,E. The letters are  ::
    :: determined by the number of 1's in a byte::  0,1,or 2 yield A,::
    :: 3 yields B, 4 yields C, 5 yields D and 6,7 or 8 yield E. Thus  ::
    :: we have a monkey at a typewriter hitting five keys with vari-  ::
    :: ous probabilities (37,56,70,56,37 over 256).  There are 5^5    ::
    :: possible 5-letter words, and from a string of 256,000 (over-   ::
    :: lapping) 5-letter words, counts are made on the frequencies    ::
    :: for each word.   The quadratic form in the weak inverse of     ::
    :: the covariance matrix of the cell counts provides a chisquare  ::
    :: test::  Q5-Q4, the difference of the naive Pearson sums of     ::
    :: (OBS-EXP)^2/EXP on counts for 5- and 4-letter cell counts.     ::
    ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
  Test results for block7.rng
 Chi-square with 5^5-5^4=2500 d.of f. for sample size:2560000
                              chisquare  equiv normal  p-value
  Results fo COUNT-THE-1's in successive bytes:
 byte stream for block7.rng       2487.17      -.181      .427992
 byte stream for block7.rng       2655.38      2.197      .986004

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

    ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
    ::     This is the COUNT-THE-1's TEST for specific bytes.         ::
    :: Consider the file under test as a stream of 32-bit integers.   ::
    :: From each integer, a specific byte is chosen , say the left-   ::
    :: most::  bits 1 to 8. Each byte can contain from 0 to 8 1's,    ::
    :: with probabilitie 1,8,28,56,70,56,28,8,1 over 256.  Now let    ::
    :: the specified bytes from successive integers provide a string  ::
    :: of (overlapping) 5-letter words, each "letter" taking values   ::
    :: A,B,C,D,E. The letters are determined  by the number of 1's,   ::
    :: in that byte::  0,1,or 2 ---> A, 3 ---> B, 4 ---> C, 5 ---> D,::
    :: and  6,7 or 8 ---> E.  Thus we have a monkey at a typewriter   ::
    :: hitting five keys with with various probabilities::  37,56,70,::
    :: 56,37 over 256. There are 5^5 possible 5-letter words, and     ::
    :: from a string of 256,000 (overlapping) 5-letter words, counts  ::
    :: are made on the frequencies for each word. The quadratic form  ::
    :: in the weak inverse of the covariance matrix of the cell       ::
    :: counts provides a chisquare test::  Q5-Q4, the difference of   ::
    :: the naive Pearson  sums of (OBS-EXP)^2/EXP on counts for 5-    ::
    :: and 4-letter cell counts.                                      ::
    ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
```

```
 Chi-square with 5^5-5^4=2500 d.of f. for sample size: 256000
                     chisquare  equiv normal  p value
  Results for COUNT-THE-1's in specified bytes:
           bits  1 to  8  2671.31      2.423      .992296
           bits  2 to  9  2530.80       .436      .668404
           bits  3 to 10  2459.45      -.573      .283184
           bits  4 to 11  2560.92       .861      .805514
           bits  5 to 12  2559.04       .835      .798121
           bits  6 to 13  2602.63      1.451      .926676
           bits  7 to 14  2359.06     -1.993      .023123
           bits  8 to 15  2525.92       .367      .643040
           bits  9 to 16  2403.80     -1.360      .086846
           bits 10 to 17  2453.50      -.658      .255385
           bits 11 to 18  2441.18      -.832      .202762
           bits 12 to 19  2569.72       .986      .837940
           bits 13 to 20  2555.82       .789      .785081
           bits 14 to 21  2416.67     -1.178      .119306
           bits 15 to 22  2665.60      2.342      .990409
           bits 16 to 23  2637.61      1.946      .974176
           bits 17 to 24  2583.10      1.175      .880045
           bits 18 to 25  2505.24       .074      .529563
           bits 19 to 26  2498.78      -.017      .493110
           bits 20 to 27  2429.44      -.998      .159179
           bits 21 to 28  2578.95      1.117      .867904
           bits 22 to 29  2422.27     -1.099      .135822
           bits 23 to 30  2560.72       .859      .804757
           bits 24 to 31  2447.31      -.745      .228083
           bits 25 to 32  2623.04      1.740      .959080

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
     ::               THIS IS A PARKING LOT TEST                       ::
     :: In a square of side 100, randomly "park" a car---a circle of  ::
     :: radius 1.   Then try to park a 2nd, a 3rd, and so on, each    ::
     :: time parking "by ear".  That is, if an attempt to park a car  ::
     :: causes a crash with one already parked, try again at a new    ::
     :: random location. (To avoid path problems, consider parking    ::
     :: helicopters rather than cars.)   Each attempt leads to either ::
     :: a crash or a success, the latter followed by an increment to  ::
     :: the list of cars already parked. If we plot n:  the number of ::
     :: attempts, versus k::  the number successfully parked, we get a::
     :: curve that should be similar to those provided by a perfect   ::
     :: random number generator.  Theory for the behavior of such a   ::
     :: random curve seems beyond reach, and as graphics displays are ::
     :: not available for this battery of tests, a simple characteriz ::
     :: ation of the random experiment is used: k, the number of cars ::
     :: successfully parked after n=12,000 attempts. Simulation shows ::
     :: that k should average 3523 with sigma 21.9 and is very close  ::
     :: to normally distributed.  Thus (k-3523)/21.9 should be a st-  ::
     :: andard normal variable, which, converted to a uniform varia-  ::
     :: ble, provides input to a KSTEST based on a sample of 10.      ::
     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
           CDPARK: result of ten tests on file block7.rng
            Of 12,000 tries, the average no. of successes
                 should be 3523 with sigma=21.9
            Successes: 3576    z-score:  2.420 p-value: .992242
```

```
   Successes: 3487    z-score: -1.644 p-value: .050105
   Successes: 3586    z-score:  2.877 p-value: .997991
   Successes: 3502    z-score:  -.959 p-value: .168804
   Successes: 3501    z-score: -1.005 p-value: .157553
   Successes: 3533    z-score:   .457 p-value: .676028
   Successes: 3502    z-score:  -.959 p-value: .168804
   Successes: 3552    z-score:  1.324 p-value: .907282
   Successes: 3555    z-score:  1.461 p-value: .928018
   Successes: 3504    z-score:  -.868 p-value: .192812

  square size   avg. no.  parked   sample sigma
    100.            3529.800       33.621
   KSTEST for the above 10: p=  .942137

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

     :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
     ::                THE MINIMUM DISTANCE TEST                        ::
     :: It does this 100 times::   choose n=8000 random points in a     ::
     :: square of side 10000.  Find d, the minimum distance between     ::
     :: the (n^2-n)/2 pairs of points.  If the points are truly inde-   ::
     :: pendent uniform, then d^2, the square of the minimum distance   ::
     :: should be (very close to) exponentially distributed with mean   ::
     :: .995 .  Thus 1-exp(-d^2/.995) should be uniform on [0,1) and    ::
     :: a KSTEST on the resulting 100 values serves as a test of uni-   ::
     :: formity for random points in the square. Test numbers=0 mod 5   ::
     :: are printed but the KSTEST is based on the full set of 100      ::
     :: random choices of 8000 points in the 10000x10000 square.        ::
     :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
               This is the MINIMUM DISTANCE test
              for random integers in the file block7.rng
     Sample no.    d^2     avg     equiv uni
           5     .6298   1.2579     .468992
          10     .4375   1.0111     .355762
          15     .2750    .9661     .241502
          20    1.8928    .8954     .850770
          25     .3661    .7976     .307844
          30     .5591    .7787     .429891
          35    1.1869    .7616     .696662
          40    2.4796    .8077     .917257
          45     .5980    .8196     .451718
          50     .7859    .8383     .546078
          55     .2071    .8181     .187945
          60     .9889    .8003     .629864
          65     .4289    .7858     .350155
          70     .7257    .7641     .517774
          75    2.2272    .7788     .893371
          80     .1968    .7566     .179480
          85     .0156    .7603     .015588
          90     .3938    .7972     .326867
          95    1.6271    .7931     .805103
         100     .1752    .8205     .161470
     MINIMUM DISTANCE TEST for block7.rng
          Result of KS test on 20 transformed mindist^2's:
                                  p-value= .739882

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
```

```
     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
     ::               THE 3DSPHERES TEST                                ::
     :: Choose  4000 random points in a cube of edge 1000.  At each     ::
     :: point, center a sphere large enough to reach the next closest ::
     :: point. Then the volume of the smallest such sphere is (very    ::
     :: close to) exponentially distributed with mean 120pi/3.  Thus  ::
     :: the radius cubed is exponential with mean 30. (The mean is     ::
     :: obtained by extensive simulation).  The 3DSPHERES test gener- ::
     :: ates 4000 such spheres 20 times.  Each min radius cubed leads ::
     :: to a uniform variable by means of 1-exp(-r^3/30.), then a      ::
     ::  KSTEST is done on the 20 p-values.                            ::
     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
              The 3DSPHERES test for file block7.rng
 sample no:  1     r^3=  37.433     p-value= .71286
 sample no:  2     r^3=    .306     p-value= .01016
 sample no:  3     r^3=  43.498     p-value= .76541
 sample no:  4     r^3=   9.718     p-value= .27669
 sample no:  5     r^3=  11.180     p-value= .31110
 sample no:  6     r^3=  48.293     p-value= .80007
 sample no:  7     r^3=  17.462     p-value= .44126
 sample no:  8     r^3=  55.525     p-value= .84290
 sample no:  9     r^3=  27.461     p-value= .59964
 sample no: 10     r^3=  62.657     p-value= .87614
 sample no: 11     r^3=   4.779     p-value= .14726
 sample no: 12     r^3=  44.232     p-value= .77109
 sample no: 13     r^3=  13.391     p-value= .36006
 sample no: 14     r^3=  16.130     p-value= .41590
 sample no: 15     r^3=  35.358     p-value= .69229
 sample no: 16     r^3=   7.551     p-value= .22253
 sample no: 17     r^3= 108.515     p-value= .97314
 sample no: 18     r^3=   6.018     p-value= .18175
 sample no: 19     r^3=  44.436     p-value= .77263
 sample no: 20     r^3=   5.681     p-value= .17253
  A KS test is applied to those 20 p-values.
---------------------------------------------------------
       3DSPHERES test for file block7.rng           p-value= .098354
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
     ::      This is the SQEEZE test                                    ::
     ::  Random integers are floated to get uniforms on [0,1). Start- ::
     ::  ing with k=2^31=2147483647, the test finds j, the number of  ::
     ::  iterations necessary to reduce k to 1, using the reduction   ::
     ::  k=ceiling(k*U), with U provided by floating integers from    ::
     ::  the file being tested.  Such j's are found 100,000 times,    ::
     ::  then counts for the number of times j was <=6,7,...,47,>=48  ::
     ::  are used to provide a chi-square test for cell frequencies.  ::
     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
            RESULTS OF SQUEEZE TEST FOR block7.rng
         Table of standardized frequency counts
     ( (obs-exp)/sqrt(exp) )^2
        for j taking values <=6,7,8,...,47,>=48:
    -1.5     1.8     -.4     -.8     1.2      .0
      .1     -.7     1.7      .9      .9      .2
    -1.1     -.4      .5     -.4     -.2     -.6
      .9    -1.0      .2     -.2      .1     -.5
```

```
  1.7     1.1      .6     -.6     -.8     -.4
   .3    -1.7     -.3    -1.7     -.9    -1.6
 -1.4    -1.7    -2.0     -.7     -.6     1.0
 -1.1
       Chi-square with 42 degrees of freedom: 43.734
          z-score=   .189  p-value= .602332
```

```
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

     :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
     ::             The  OVERLAPPING SUMS test                         ::
     :: Integers are floated to get a sequence U(1),U(2),... of uni-  ::
     :: form [0,1) variables.  Then overlapping sums,                  ::
     ::   S(1)=U(1)+...+U(100), S2=U(2)+...+U(101),... are formed.    ::
     :: The S's are virtually normal with a certain covariance mat-   ::
     :: rix.  A linear transformation of the S's converts them to a   ::
     :: sequence of independent standard normals, which are converted ::
     :: to uniform variables for a KSTEST. The  p-values from ten     ::
     :: KSTESTs are given still another KSTEST.                       ::
     :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
                Test no.  1      p-value  .676187
                Test no.  2      p-value  .186294
                Test no.  3      p-value  .022401
                Test no.  4      p-value  .617651
                Test no.  5      p-value  .544124
                Test no.  6      p-value  .302474
                Test no.  7      p-value  .557463
                Test no.  8      p-value  .802135
                Test no.  9      p-value  .411520
                Test no. 10      p-value  .563264
   Results of the OSUM test for block7.rng
        KSTEST on the above 10 p-values:  .378851

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

     :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
     ::     This is the RUNS test.  It counts runs up, and runs down, ::
     :: in a sequence of uniform [0,1) variables, obtained by float-  ::
     :: ing the 32-bit integers in the specified file. This example   ::
     :: shows how runs are counted:  .123,.357,.789,.425,.224,.416,.95::
     :: contains an up-run of length 3, a down-run of length 2 and an ::
     :: up-run of (at least) 2, depending on the next values.  The    ::
     :: covariance matrices for the runs-up and runs-down are well    ::
     :: known, leading to chisquare tests for quadratic forms in the  ::
     :: weak inverses of the covariance matrices.  Runs are counted   ::
     :: for sequences of length 10,000.  This is done ten times. Then ::
     :: repeated.                                                     ::
     :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
           The RUNS test for file block7.rng
     Up and down runs in a sample of 10000
```

```
                 Run test for block7.rng     :
       runs up; ks test for 10 p's: .096866
     runs down; ks test for 10 p's: .802117
                 Run test for block7.rng     :
       runs up; ks test for 10 p's: .563537
```

```
     runs down; ks test for 10 p's: .400210

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

     :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
     :: This is the CRAPS TEST. It plays 200,000 games of craps, finds::
     :: the number of wins and the number of throws necessary to end  ::
     :: each game.  The number of wins should be (very close to) a    ::
     :: normal with mean 200000p and variance 200000p(1-p), with      ::
     :: p=244/495.  Throws necessary to complete the game can vary    ::
     :: from 1 to infinity, but counts for all>21 are lumped with 21. ::
     :: A chi-square test is made on the no.-of-throws cell counts.   ::
     :: Each 32-bit integer from the test file provides the value for ::
     :: the throw of a die, by floating to [0,1), multiplying by 6    ::
     :: and taking 1 plus the integer part of the result.            ::
     :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
                Results of craps test for block7.rng
   No. of wins:  Observed Expected
                                98425    98585.86
                  98425= No. of wins, z-score= -.719 pvalue= .23593
    Analysis of Throws-per-Game:
  Chisq=  17.64 for 20 degrees of freedom, p=  .38884
                 Throws Observed Expected  Chisq     Sum
                    1    66636    66666.7    .014     .014
                    2    37642    37654.3    .004     .018
                    3    27016    26954.7    .139     .157
                    4    19231    19313.5    .352     .509
                    5    14029    13851.4   2.277    2.786
                    6     9950     9943.5    .004    2.790
                    7     7061     7145.0    .988    3.778
                    8     5032     5139.1   2.231    6.009
                    9     3743     3699.9    .503    6.512
                   10     2708     2666.3    .652    7.164
                   11     1873     1923.3   1.317    8.481
                   12     1394     1388.7    .020    8.501
                   13     1009     1003.7    .028    8.529
                   14      740      726.1    .265    8.794
                   15      490      525.8   2.442   11.236
                   16      405      381.2   1.492   12.728
                   17      280      276.5    .043   12.772
                   18      216      200.8   1.146   13.917
                   19      141      146.0    .170   14.088
                   20      125      106.2   3.322   17.410
                   21      279      287.1    .229   17.639
              SUMMARY  FOR block7.rng
                p-value for no. of wins: .235930
                p-value for throws/game: .388841

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

  Results of DIEHARD battery of tests sent to file report7.txt



        NOTE: Most of the tests in DIEHARD return a p-value, which
        should be uniform on [0,1) if the input file contains truly
```

```
    independent random bits.   Those p-values are obtained by
    p=F(X), where F is the assumed distribution of the sample
    random variable X---often normal. But that assumed F is just
    an asymptotic approximation, for which the fit will be worst
    in the tails. Thus you should not be surprised with
    occasional p-values near 0 or 1, such as .0012 or .9983.
    When a bit stream really FAILS BIG, you will get p's of 0 or
    1 to six or more places.  By all means, do not, as a
    Statistician might, think that a p < .025 or p> .975 means
    that the RNG has "failed the test at the .05 level".  Such
    p's happen among the hundreds that DIEHARD produces, even
    with good RNG's.  So keep in mind that " p happens".
  ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
  ::            This is the BIRTHDAY SPACINGS TEST                ::
  :: Choose m birthdays in a year of n days.  List the spacings   ::
  :: between the birthdays.  If j is the number of values that    ::
  :: occur more than once in that list, then j is asymptotically  ::
  :: Poisson distributed with mean m^3/(4n).  Experience shows n  ::
  :: must be quite large, say n>=2^18, for comparing the results  ::
  :: to the Poisson distribution with that mean.  This test uses  ::
  :: n=2^24 and m=2^9,  so that the underlying distribution for j ::
  :: is taken to be Poisson with lambda=2^27/(2^26)=2.  A sample  ::
  :: of 500 j's is taken, and a chi-square goodness of fit test   ::
  :: provides a p value.  The first test uses bits 1-24 (counting ::
  :: from the left) from integers in the specified file.          ::
  ::   Then the file is closed and reopened. Next, bits 2-25 are  ::
  :: used to provide birthdays, then 3-26 and so on to bits 9-32. ::
  :: Each set of bits provides a p-value, and the nine p-values   ::
  :: provide a sample for a KSTEST.                               ::
  ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
BIRTHDAY SPACINGS TEST, M= 512 N=2**24 LAMBDA=  2.0000
          Results for BLOCKX.RNG
                  For a sample of size 500:     mean
          BLOCKX.RNG      using bits  1 to 24   1.888
 duplicate       number       number
 spacings       observed     expected
       0          74.       67.668
       1         162.      135.335
       2         122.      135.335
       3          75.       90.224
       4          36.       45.112
       5          21.       18.045
 6 to INF         10.        8.282
Chisquare with  6 d.o.f. =    12.41 p-value=  .946574
 :::::::::::::::::::::::::::::::::::::::::
                  For a sample of size 500:     mean
          BLOCKX.RNG      using bits  2 to 25   1.990
 duplicate       number       number
 spacings       observed     expected
       0          66.       67.668
       1         138.      135.335
       2         137.      135.335
       3          95.       90.224
       4          35.       45.112
       5          18.       18.045
 6 to INF         11.        8.282
Chisquare with  6 d.o.f. =     3.53 p-value=  .259454
```

```
 :::::::::::::::::::::::::::::::::::::::::
                  For a sample of size 500:     mean
          BLOCKX.RNG      using bits  3 to 26   1.980
 duplicate       number       number
 spacings       observed     expected
      0          74.       67.668
      1         131.      135.335
      2         141.      135.335
      3          81.       90.224
      4          42.       45.112
      5          22.       18.045
 6 to INF         9.        8.282
Chisquare with  6 d.o.f. =     3.06 p-value=  .198136
 :::::::::::::::::::::::::::::::::::::::::
                  For a sample of size 500:     mean
          BLOCKX.RNG      using bits  4 to 27   2.106
 duplicate       number       number
 spacings       observed     expected
      0          61.       67.668
      1         129.      135.335
      2         136.      135.335
      3          92.       90.224
      4          46.       45.112
      5          24.       18.045
 6 to INF        12.        8.282
Chisquare with  6 d.o.f. =     4.64 p-value=  .409792
 :::::::::::::::::::::::::::::::::::::::::
                  For a sample of size 500:     mean
          BLOCKX.RNG      using bits  5 to 28   1.944
 duplicate       number       number
 spacings       observed     expected
      0          73.       67.668
      1         135.      135.335
      2         142.      135.335
      3          77.       90.224
      4          50.       45.112
      5          17.       18.045
 6 to INF         6.        8.282
Chisquare with  6 d.o.f. =     3.91 p-value=  .310630
 :::::::::::::::::::::::::::::::::::::::::
                  For a sample of size 500:     mean
          BLOCKX.RNG      using bits  6 to 29   1.948
 duplicate       number       number
 spacings       observed     expected
      0          86.       67.668
      1         125.      135.335
      2         133.      135.335
      3          76.       90.224
      4          53.       45.112
      5          22.       18.045
 6 to INF         5.        8.282
Chisquare with  6 d.o.f. =    11.59 p-value=  .928112
 :::::::::::::::::::::::::::::::::::::::::
                  For a sample of size 500:     mean
          BLOCKX.RNG      using bits  7 to 30   1.964
 duplicate       number       number
 spacings       observed     expected
```

```
       0          74.       67.668
       1         131.      135.335
       2         134.      135.335
       3          92.       90.224
       4          46.       45.112
       5          16.       18.045
  6 to INF         7.        8.282
 Chisquare with  6 d.o.f. =     1.23 p-value=  .024479
  :::::::::::::::::::::::::::::::::::::::::
                  For a sample of size 500:     mean
          BLOCKX.RNG      using bits  8 to 31   2.094
  duplicate       number       number
  spacings       observed     expected
       0          57.       67.668
       1         122.      135.335
       2         142.      135.335
       3         113.       90.224
       4          40.       45.112
       5          16.       18.045
  6 to INF        10.        8.282
 Chisquare with  6 d.o.f. =    10.24 p-value=  .885142
  :::::::::::::::::::::::::::::::::::::::::
                  For a sample of size 500:     mean
          BLOCKX.RNG      using bits  9 to 32   1.956
  duplicate       number       number
  spacings       observed     expected
       0          62.       67.668
       1         145.      135.335
       2         142.      135.335
       3          87.       90.224
       4          42.       45.112
       5          14.       18.045
  6 to INF         8.        8.282
 Chisquare with  6 d.o.f. =     2.74 p-value=  .159196
  :::::::::::::::::::::::::::::::::::::::::
   The 9 p-values were
        .946574   .259454   .198136   .409792   .310630
        .928112   .024479   .885142   .159196
  A KSTEST for the 9 p-values yields  .551415

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
     ::            THE OVERLAPPING 5-PERMUTATION TEST                ::
     :: This is the OPERM5 test.  It looks at a sequence of one mill- ::
     :: ion 32-bit random integers.  Each set of five consecutive     ::
     :: integers can be in one of 120 states, for the 5! possible or- ::
     :: derings of five numbers.  Thus the 5th, 6th, 7th,...numbers   ::
     :: each provide a state. As many thousands of state transitions  ::
     :: are observed,  cumulative counts are made of the number of    ::
     :: occurences of each state.  Then the quadratic form in the     ::
     :: weak inverse of the 120x120 covariance matrix yields a test   ::
     :: equivalent to the likelihood ratio test that the 120 cell     ::
     :: counts came from the specified (asymptotically) normal dis-   ::
     :: tribution with the specified 120x120 covariance matrix (with  ::
     :: rank 99).  This version uses 1,000,000 integers, twice.       ::
     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
```

```
          OPERM5 test for file BLOCKX.RNG
     For a sample of 1,000,000 consecutive 5-tuples,
 chisquare for 99 degrees of freedom=100.986; p-value= .574389
          OPERM5 test for file BLOCKX.RNG
     For a sample of 1,000,000 consecutive 5-tuples,
 chisquare for 99 degrees of freedom= 87.084; p-value= .201601
     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
     :: This is the BINARY RANK TEST for 31x31 matrices. The leftmost ::
     :: 31 bits of 31 random integers from the test sequence are used ::
     :: to form a 31x31 binary matrix over the field {0,1}. The rank  ::
     :: is determined. That rank can be from 0 to 31, but ranks< 28   ::
     :: are rare, and their counts are pooled with those for rank 28. ::
     :: Ranks are found for 40,000 such random matrices and a chisqua-::
     :: re test is performed on counts for ranks 31,30,29 and <=28.   ::
     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
    Binary rank test for BLOCKX.RNG
         Rank test for 31x31 binary matrices:
        rows from leftmost 31 bits of each 32-bit integer
      rank   observed  expected (o-e)^2/e  sum
        28       216     211.4   .099304     .099
        29      5174    5134.0   .311487     .411
        30     23171   23103.0   .199871     .611
        31     11439   11551.5  1.096110    1.707
  chisquare= 1.707 for 3 d. of f.; p-value= .460274
--------------------------------------------------------------
     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
     :: This is the BINARY RANK TEST for 32x32 matrices. A random 32x ::
     :: 32 binary matrix is formed, each row a 32-bit random integer. ::
     :: The rank is determined. That rank can be from 0 to 32, ranks  ::
     :: less than 29 are rare, and their counts are pooled with those ::
     :: for rank 29.  Ranks are found for 40,000 such random matrices ::
     :: and a chisquare test is performed on counts for ranks  32,31, ::
     :: 30 and <=29.                                                 ::
     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
    Binary rank test for BLOCKX.RNG
         Rank test for 32x32 binary matrices:
        rows from leftmost 32 bits of each 32-bit integer
      rank   observed  expected (o-e)^2/e  sum
        29       223     211.4   .634489     .634
        30      5177    5134.0   .359976     .994
        31     23165   23103.0   .166133    1.161
        32     11435   11551.5  1.175424    2.336
  chisquare= 2.336 for 3 d. of f.; p-value= .557551
--------------------------------------------------------------

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
     :: This is the BINARY RANK TEST for 6x8 matrices.  From each of  ::
     :: six random 32-bit integers from the generator under test, a   ::
     :: specified byte is chosen, and the resulting six bytes form a  ::
     :: 6x8 binary matrix whose rank is determined.  That rank can be ::
     :: from 0 to 6, but ranks 0,1,2,3 are rare; their counts are     ::
     :: pooled with those for rank 4. Ranks are found for 100,000     ::
     :: random matrices, and a chi-square test is performed on       ::
     :: counts for ranks 6,5 and <=4.                                ::
     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
```

```
    Binary Rank Test for BLOCKX.RNG
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG BLOCKX.RNG
b-rank test for bits  1 to  8
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          915        944.3         .909        .909
     r =5        21833      21743.9         .365       1.274
     r =6        77252      77311.8         .046       1.321
                   p=1-exp(-SUM/2)= .48330
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG BLOCKX.RNG
b-rank test for bits  2 to  9
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          924        944.3         .436        .436
     r =5        21728      21743.9         .012        .448
     r =6        77348      77311.8         .017        .465
                   p=1-exp(-SUM/2)= .20746
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG BLOCKX.RNG
b-rank test for bits  3 to 10
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          891        944.3        3.009       3.009
     r =5        21443      21743.9        4.164       7.173
     r =6        77666      77311.8        1.623       8.795
                   p=1-exp(-SUM/2)= .98769
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG BLOCKX.RNG
b-rank test for bits  4 to 11
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          928        944.3         .281        .281
     r =5        21523      21743.9        2.244       2.526
     r =6        77549      77311.8         .728       3.253
                   p=1-exp(-SUM/2)= .80341
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG BLOCKX.RNG
b-rank test for bits  5 to 12
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          872        944.3        5.536       5.536
     r =5        21476      21743.9        3.301       8.837
     r =6        77652      77311.8        1.497      10.333
                   p=1-exp(-SUM/2)= .99430
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG BLOCKX.RNG
b-rank test for bits  6 to 13
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          957        944.3         .171        .171
     r =5        21744      21743.9         .000        .171
     r =6        77299      77311.8         .002        .173
                   p=1-exp(-SUM/2)= .08281
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG BLOCKX.RNG
b-rank test for bits  7 to 14
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4         1008        944.3        4.297       4.297
     r =5        21606      21743.9         .875       5.171
     r =6        77386      77311.8         .071       5.243
                   p=1-exp(-SUM/2)= .92729
```

```
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG BLOCKX.RNG
b-rank test for bits  8 to 15
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4         939        944.3         .030         .030
     r =5       21850      21743.9         .518         .547
     r =6       77211      77311.8         .131         .679
                   p=1-exp(-SUM/2)= .28784
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG BLOCKX.RNG
b-rank test for bits  9 to 16
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4         940        944.3         .020         .020
     r =5       22067      21743.9        4.801        4.821
     r =6       76993      77311.8        1.315        6.135
                   p=1-exp(-SUM/2)= .95347
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG BLOCKX.RNG
b-rank test for bits 10 to 17
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4         932        944.3         .160         .160
     r =5       21568      21743.9        1.423        1.583
     r =6       77500      77311.8         .458        2.041
                   p=1-exp(-SUM/2)= .63964
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG BLOCKX.RNG
b-rank test for bits 11 to 18
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4         910        944.3        1.246        1.246
     r =5       21717      21743.9         .033        1.279
     r =6       77373      77311.8         .048        1.328
                   p=1-exp(-SUM/2)= .48513
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG BLOCKX.RNG
b-rank test for bits 12 to 19
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4         925        944.3         .395         .395
     r =5       21645      21743.9         .450         .844
     r =6       77430      77311.8         .181        1.025
                   p=1-exp(-SUM/2)= .40102
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG BLOCKX.RNG
b-rank test for bits 13 to 20
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4         976        944.3        1.064        1.064
     r =5       21695      21743.9         .110        1.174
     r =6       77329      77311.8         .004        1.178
                   p=1-exp(-SUM/2)= .44508
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG BLOCKX.RNG
b-rank test for bits 14 to 21
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4         924        944.3         .436         .436
     r =5       21800      21743.9         .145         .581
     r =6       77276      77311.8         .017         .598
                   p=1-exp(-SUM/2)= .25836
   Rank of a 6x8 binary matrix,
```

```
rows formed from eight bits of the RNG BLOCKX.RNG
b-rank test for bits 15 to 22
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          923       944.3        .481         .481
     r =5        21728     21743.9        .012         .492
     r =6        77349     77311.8        .018         .510
                   p=1-exp(-SUM/2)= .22510
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG BLOCKX.RNG
b-rank test for bits 16 to 23
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          928       944.3        .281         .281
     r =5        21924     21743.9       1.492        1.773
     r =6        77148     77311.8        .347        2.120
                   p=1-exp(-SUM/2)= .65358
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG BLOCKX.RNG
b-rank test for bits 17 to 24
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          921       944.3        .575         .575
     r =5        21812     21743.9        .213         .788
     r =6        77267     77311.8        .026         .814
                   p=1-exp(-SUM/2)= .33443
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG BLOCKX.RNG
b-rank test for bits 18 to 25
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          938       944.3        .042         .042
     r =5        21776     21743.9        .047         .089
     r =6        77286     77311.8        .009         .098
                   p=1-exp(-SUM/2)= .04784
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG BLOCKX.RNG
b-rank test for bits 19 to 26
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          910       944.3       1.246        1.246
     r =5        21733     21743.9        .005        1.251
     r =6        77357     77311.8        .026        1.278
                   p=1-exp(-SUM/2)= .47214
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG BLOCKX.RNG
b-rank test for bits 20 to 27
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          884       944.3       3.851        3.851
     r =5        21776     21743.9        .047        3.898
     r =6        77340     77311.8        .010        3.908
                   p=1-exp(-SUM/2)= .85832
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG BLOCKX.RNG
b-rank test for bits 21 to 28
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          928       944.3        .281         .281
     r =5        21661     21743.9        .316         .597
     r =6        77411     77311.8        .127         .725
                   p=1-exp(-SUM/2)= .30398
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG BLOCKX.RNG
```

```
b-rank test for bits 22 to 29
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          930       944.3         .217        .217
     r =5        21498     21743.9        2.781       2.997
     r =6        77572     77311.8         .876       3.873
                   p=1-exp(-SUM/2)= .85580
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG BLOCKX.RNG
b-rank test for bits 23 to 30
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          950       944.3         .034        .034
     r =5        21693     21743.9         .119        .154
     r =6        77357     77311.8         .026        .180
                   p=1-exp(-SUM/2)= .08605
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG BLOCKX.RNG
b-rank test for bits 24 to 31
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          950       944.3         .034        .034
     r =5        21769     21743.9         .029        .063
     r =6        77281     77311.8         .012        .076
                   p=1-exp(-SUM/2)= .03711
   Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG BLOCKX.RNG
b-rank test for bits 25 to 32
                OBSERVED   EXPECTED     (O-E)^2/E      SUM
     r<=4          961       944.3         .295        .295
     r =5        21728     21743.9         .012        .307
     r =6        77311     77311.8         .000        .307
                   p=1-exp(-SUM/2)= .14227
  TEST SUMMARY, 25 tests on 100,000 random 6x8 matrices
These should be 25 uniform [0,1] random variables:
    .483296     .207460     .987694     .803413     .994297
    .082814     .927293     .287842     .953469     .639645
    .485133     .401019     .445083     .258356     .225095
    .653577     .334429     .047842     .472143     .858322
    .303976     .855804     .086053     .037113     .142270
  brank test summary for BLOCKX.RNG
      The KS test for those 25 supposed UNI's yields
                 KS p-value= .396534

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

     :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
     ::                   THE BITSTREAM TEST                          ::
     :: The file under test is viewed as a stream of bits. Call them  ::
     :: b1,b2,... .  Consider an alphabet with two "letters", 0 and 1 ::
     :: and think of the stream of bits as a succession of 20-letter  ::
     :: "words", overlapping.  Thus the first word is b1b2...b20, the ::
     :: second is b2b3...b21, and so on.  The bitstream test counts   ::
     :: the number of missing 20-letter (20-bit) words in a string of ::
     :: 2^21 overlapping 20-letter words.  There are 2^20 possible 20 ::
     :: letter words.  For a truly random string of 2^21+19 bits, the ::
     :: number of missing words j should be (very close to) normally  ::
     :: distributed with mean 141,909 and sigma 428.  Thus            ::
     ::  (j-141909)/428 should be a standard normal variate (z score) ::
     :: that leads to a uniform [0,1) p value.  The test is repeated  ::
```

```
     :: twenty times.                                                 ::
     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
 THE OVERLAPPING 20-tuples BITSTREAM  TEST, 20 BITS PER WORD, N words
   This test uses N=2^21 and samples the bitstream 20 times.
  No. missing words should average  141909. with sigma=428.
---------------------------------------------------------
 tst no  1:  141822 missing words,    -.20 sigmas from mean, p-value= .41916
 tst no  2:  142480 missing words,    1.33 sigmas from mean, p-value= .90879
 tst no  3:  141369 missing words,   -1.26 sigmas from mean, p-value= .10339
 tst no  4:  142381 missing words,    1.10 sigmas from mean, p-value= .86478
 tst no  5:  141364 missing words,   -1.27 sigmas from mean, p-value= .10131
 tst no  6:  142052 missing words,     .33 sigmas from mean, p-value= .63056
 tst no  7:  141954 missing words,     .10 sigmas from mean, p-value= .54156
 tst no  8:  141624 missing words,    -.67 sigmas from mean, p-value= .25250
 tst no  9:  142049 missing words,     .33 sigmas from mean, p-value= .62791
 tst no 10:  141866 missing words,    -.10 sigmas from mean, p-value= .45968
 tst no 11:  142330 missing words,     .98 sigmas from mean, p-value= .83717
 tst no 12:  141971 missing words,     .14 sigmas from mean, p-value= .55729
 tst no 13:  141785 missing words,    -.29 sigmas from mean, p-value= .38572
 tst no 14:  141932 missing words,     .05 sigmas from mean, p-value= .52112
 tst no 15:  141778 missing words,    -.31 sigmas from mean, p-value= .37948
 tst no 16:  142086 missing words,     .41 sigmas from mean, p-value= .66012
 tst no 17:  142203 missing words,     .69 sigmas from mean, p-value= .75369
 tst no 18:  142071 missing words,     .38 sigmas from mean, p-value= .64719
 tst no 19:  142064 missing words,     .36 sigmas from mean, p-value= .64109
 tst no 20:  141596 missing words,    -.73 sigmas from mean, p-value= .23206

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
     ::             The tests OPSO, OQSO and DNA                      ::
     ::         OPSO means Overlapping-Pairs-Sparse-Occupancy         ::
     :: The OPSO test considers 2-letter words from an alphabet of    ::
     :: 1024 letters.  Each letter is determined by a specified ten   ::
     :: bits from a 32-bit integer in the sequence to be tested. OPSO ::
     :: generates  2^21 (overlapping) 2-letter words  (from 2^21+1    ::
     :: "keystrokes")  and counts the number of missing words---that  ::
     :: is 2-letter words which do not appear in the entire sequence. ::
     :: That count should be very close to normally distributed with  ::
     :: mean 141,909, sigma 290. Thus (missingwrds-141909)/290 should ::
     :: be a standard normal variable. The OPSO test takes 32 bits at ::
     :: a time from the test file and uses a designated set of ten    ::
     :: consecutive bits. It then restarts the file for the next de-  ::
     :: signated 10 bits, and so on.                                  ::
     ::                                                               ::
     ::     OQSO means Overlapping-Quadruples-Sparse-Occupancy        ::
     ::   The test OQSO is similar, except that it considers 4-letter ::
     :: words from an alphabet of 32 letters, each letter determined  ::
     :: by a designated string of 5 consecutive bits from the test    ::
     :: file, elements of which are assumed 32-bit random integers.   ::
     :: The mean number of missing words in a sequence of 2^21 four-  ::
     :: letter words,  (2^21+3 "keystrokes"), is again 141909, with   ::
     :: sigma = 295.  The mean is based on theory; sigma comes from   ::
     :: extensive simulation.                                         ::
     ::                                                               ::
     ::    The DNA test considers an alphabet of 4 letters::  C,G,A,T,::
     :: determined by two designated bits in the sequence of random   ::
```

```
    :: integers being tested.  It considers 10-letter words, so that ::
    :: as in OPSO and OQSO, there are 2^20 possible words, and the   ::
    :: mean number of missing words from a string of 2^21  (over-    ::
    :: lapping)  10-letter  words (2^21+9 "keystrokes") is 141909.   ::
    :: The standard deviation sigma=339 was determined as for OQSO   ::
    :: by simulation.  (Sigma for OPSO, 290, is the true value (to   ::
    :: three places), not determined by simulation.                  ::
    :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
OPSO test for generator BLOCKX.RNG
 Output: No. missing words (mw), equiv normal variate (z), p-value (p)
                                                           mw     z     p
   OPSO for BLOCKX.RNG      using bits 23 to 32         141947   .130  .5517
   OPSO for BLOCKX.RNG      using bits 22 to 31         142084   .602  .7265
   OPSO for BLOCKX.RNG      using bits 21 to 30         141897  -.043  .4830
   OPSO for BLOCKX.RNG      using bits 20 to 29         142261  1.213  .8874
   OPSO for BLOCKX.RNG      using bits 19 to 28         142267  1.233  .8913
   OPSO for BLOCKX.RNG      using bits 18 to 27         142421  1.764  .9612
   OPSO for BLOCKX.RNG      using bits 17 to 26         141804  -.363  .3582
   OPSO for BLOCKX.RNG      using bits 16 to 25         141675  -.808  .2095
   OPSO for BLOCKX.RNG      using bits 15 to 24         141330 -1.998  .0229
   OPSO for BLOCKX.RNG      using bits 14 to 23         141832  -.267  .3949
   OPSO for BLOCKX.RNG      using bits 13 to 22         142663  2.599  .9953
   OPSO for BLOCKX.RNG      using bits 12 to 21         141862  -.163  .4352
   OPSO for BLOCKX.RNG      using bits 11 to 20         142035   .433  .6676
   OPSO for BLOCKX.RNG      using bits 10 to 19         141861  -.167  .4338
   OPSO for BLOCKX.RNG      using bits  9 to 18         141900  -.032  .4872
   OPSO for BLOCKX.RNG      using bits  8 to 17         142008   .340  .6332
   OPSO for BLOCKX.RNG      using bits  7 to 16         141849  -.208  .4176
   OPSO for BLOCKX.RNG      using bits  6 to 15         141940   .106  .5421
   OPSO for BLOCKX.RNG      using bits  5 to 14         142283  1.289  .9012
   OPSO for BLOCKX.RNG      using bits  4 to 13         142048   .478  .6837
   OPSO for BLOCKX.RNG      using bits  3 to 12         141985   .261  .6029
   OPSO for BLOCKX.RNG      using bits  2 to 11         142502  2.044  .9795
   OPSO for BLOCKX.RNG      using bits  1 to 10         142461  1.902  .9714
OQSO test for generator BLOCKX.RNG
 Output: No. missing words (mw), equiv normal variate (z), p-value (p)
                                                           mw     z     p
   OQSO for BLOCKX.RNG      using bits 28 to 32         141916   .023  .5090
   OQSO for BLOCKX.RNG      using bits 27 to 31         141831  -.266  .3953
   OQSO for BLOCKX.RNG      using bits 26 to 30         141878  -.106  .4577
   OQSO for BLOCKX.RNG      using bits 25 to 29         141984   .253  .5999
   OQSO for BLOCKX.RNG      using bits 24 to 28         142051   .480  .6845
   OQSO for BLOCKX.RNG      using bits 23 to 27         142227  1.077  .8592
   OQSO for BLOCKX.RNG      using bits 22 to 26         142245  1.138  .8724
   OQSO for BLOCKX.RNG      using bits 21 to 25         141623  -.971  .1659
   OQSO for BLOCKX.RNG      using bits 20 to 24         141753  -.530  .2981
   OQSO for BLOCKX.RNG      using bits 19 to 23         141926   .057  .5225
   OQSO for BLOCKX.RNG      using bits 18 to 22         142284  1.270  .8980
   OQSO for BLOCKX.RNG      using bits 17 to 21         141982   .246  .5973
   OQSO for BLOCKX.RNG      using bits 16 to 20         141526 -1.299  .0969
   OQSO for BLOCKX.RNG      using bits 15 to 19         141837  -.245  .4032
   OQSO for BLOCKX.RNG      using bits 14 to 18         141976   .226  .5894
   OQSO for BLOCKX.RNG      using bits 13 to 17         141937   .094  .5374
   OQSO for BLOCKX.RNG      using bits 12 to 16         142048   .470  .6808
   OQSO for BLOCKX.RNG      using bits 11 to 15         142120   .714  .7624
   OQSO for BLOCKX.RNG      using bits 10 to 14         141810  -.337  .3682
   OQSO for BLOCKX.RNG      using bits  9 to 13         141709  -.679  .2485
```

```
   OQSO for BLOCKX.RNG      using bits  8 to 12        142069   .541  .7058
   OQSO for BLOCKX.RNG      using bits  7 to 11        141930   .070  .5279
   OQSO for BLOCKX.RNG      using bits  6 to 10        142300  1.324  .9073
   OQSO for BLOCKX.RNG      using bits  5 to  9        141768  -.479  .3159
   OQSO for BLOCKX.RNG      using bits  4 to  8        142161   .853  .8032
   OQSO for BLOCKX.RNG      using bits  3 to  7        142234  1.101  .8645
   OQSO for BLOCKX.RNG      using bits  2 to  6        142048   .470  .6808
   OQSO for BLOCKX.RNG      using bits  1 to  5        141934   .084  .5333
 DNA test for generator BLOCKX.RNG
 Output: No. missing words (mw), equiv normal variate (z), p-value (p)
                                                          mw     z     p
    DNA for BLOCKX.RNG      using bits 31 to 32        141759  -.443  .3287
    DNA for BLOCKX.RNG      using bits 30 to 31        142159   .736  .7693
    DNA for BLOCKX.RNG      using bits 29 to 30        141374 -1.579  .0572
    DNA for BLOCKX.RNG      using bits 28 to 29        142117   .613  .7299
    DNA for BLOCKX.RNG      using bits 27 to 28        141506 -1.190  .1171
    DNA for BLOCKX.RNG      using bits 26 to 27        142022   .332  .6302
    DNA for BLOCKX.RNG      using bits 25 to 26        141802  -.317  .3758
    DNA for BLOCKX.RNG      using bits 24 to 25        141749  -.473  .3181
    DNA for BLOCKX.RNG      using bits 23 to 24        142024   .338  .6324
    DNA for BLOCKX.RNG      using bits 22 to 23        142114   .604  .7270
    DNA for BLOCKX.RNG      using bits 21 to 22        141733  -.520  .3015
    DNA for BLOCKX.RNG      using bits 20 to 21        142277  1.085  .8609
    DNA for BLOCKX.RNG      using bits 19 to 20        142187   .819  .7936
    DNA for BLOCKX.RNG      using bits 18 to 19        141506 -1.190  .1171
    DNA for BLOCKX.RNG      using bits 17 to 18        142206   .875  .8093
    DNA for BLOCKX.RNG      using bits 16 to 17        142153   .719  .7639
    DNA for BLOCKX.RNG      using bits 15 to 16        141699  -.620  .2675
    DNA for BLOCKX.RNG      using bits 14 to 15        141958   .144  .5571
    DNA for BLOCKX.RNG      using bits 13 to 14        141513 -1.169  .1212
    DNA for BLOCKX.RNG      using bits 12 to 13        141283 -1.848  .0323
    DNA for BLOCKX.RNG      using bits 11 to 12        141903  -.019  .4926
    DNA for BLOCKX.RNG      using bits 10 to 11        141716  -.570  .2842
    DNA for BLOCKX.RNG      using bits  9 to 10        142005   .282  .6111
    DNA for BLOCKX.RNG      using bits  8 to  9        142482  1.689  .9544
    DNA for BLOCKX.RNG      using bits  7 to  8        141934   .073  .5290
    DNA for BLOCKX.RNG      using bits  6 to  7        142024   .338  .6324
    DNA for BLOCKX.RNG      using bits  5 to  6        141677  -.685  .2466
    DNA for BLOCKX.RNG      using bits  4 to  5        141757  -.449  .3266
    DNA for BLOCKX.RNG      using bits  3 to  4        141653  -.756  .2248
    DNA for BLOCKX.RNG      using bits  2 to  3        141949   .117  .5466
    DNA for BLOCKX.RNG      using bits  1 to  2        141706  -.600  .2743

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

    ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
    ::     This is the COUNT-THE-1's TEST on a stream of bytes.      ::
    :: Consider the file under test as a stream of bytes (four per   ::
    :: 32 bit integer).  Each byte can contain from 0 to 8 1's,      ::
    :: with probabilities 1,8,28,56,70,56,28,8,1 over 256.  Now let  ::
    :: the stream of bytes provide a string of overlapping  5-letter ::
    :: words, each "letter" taking values A,B,C,D,E. The letters are ::
    :: determined by the number of 1's in a byte::  0,1,or 2 yield A,::
    :: 3 yields B, 4 yields C, 5 yields D and 6,7 or 8 yield E. Thus ::
    :: we have a monkey at a typewriter hitting five keys with vari- ::
    :: ous probabilities (37,56,70,56,37 over 256).  There are 5^5   ::
    :: possible 5-letter words, and from a string of 256,000 (over-  ::
```

```
     :: lapping) 5-letter words, counts are made on the frequencies   ::
     :: for each word.   The quadratic form in the weak inverse of    ::
     :: the covariance matrix of the cell counts provides a chisquare ::
     :: test::  Q5-Q4, the difference of the naive Pearson sums of    ::
     :: (OBS-EXP)^2/EXP on counts for 5- and 4-letter cell counts.    ::
     :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
   Test results for BLOCKX.RNG
 Chi-square with 5^5-5^4=2500 d.of f. for sample size:2560000
                              chisquare  equiv normal  p-value
  Results fo COUNT-THE-1's in successive bytes:
 byte stream for BLOCKX.RNG       2583.70      1.184      .881726
 byte stream for BLOCKX.RNG       2526.01       .368      .643523

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

     :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
     ::     This is the COUNT-THE-1's TEST for specific bytes.        ::
     :: Consider the file under test as a stream of 32-bit integers.  ::
     :: From each integer, a specific byte is chosen , say the left-  ::
     :: most::  bits 1 to 8. Each byte can contain from 0 to 8 1's,   ::
     :: with probabilitie 1,8,28,56,70,56,28,8,1 over 256.  Now let   ::
     :: the specified bytes from successive integers provide a string ::
     :: of (overlapping) 5-letter words, each "letter" taking values  ::
     :: A,B,C,D,E. The letters are determined  by the number of 1's,  ::
     :: in that byte::  0,1,or 2 ---> A, 3 ---> B, 4 ---> C, 5 ---> D,::
     :: and  6,7 or 8 ---> E.  Thus we have a monkey at a typewriter  ::
     :: hitting five keys with with various probabilities::  37,56,70,::
     :: 56,37 over 256. There are 5^5 possible 5-letter words, and    ::
     :: from a string of 256,000 (overlapping) 5-letter words, counts ::
     :: are made on the frequencies for each word. The quadratic form ::
     :: in the weak inverse of the covariance matrix of the cell      ::
     :: counts provides a chisquare test::  Q5-Q4, the difference of  ::
     :: the naive Pearson  sums of (OBS-EXP)^2/EXP on counts for 5-   ::
     :: and 4-letter cell counts.                                     ::
     :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
 Chi-square with 5^5-5^4=2500 d.of f. for sample size: 256000
                      chisquare  equiv normal  p value
  Results for COUNT-THE-1's in specified bytes:
           bits  1 to  8  2502.04       .029      .511533
           bits  2 to  9  2543.23       .611      .729498
           bits  3 to 10  2598.95      1.399      .919139
           bits  4 to 11  2462.71      -.527      .298976
           bits  5 to 12  2589.75      1.269      .897813
           bits  6 to 13  2507.29       .103      .541030
           bits  7 to 14  2499.43      -.008      .496767
           bits  8 to 15  2438.49      -.870      .192185
           bits  9 to 16  2461.94      -.538      .295211
           bits 10 to 17  2671.80      2.430      .992443
           bits 11 to 18  2581.58      1.154      .875694
           bits 12 to 19  2528.57       .404      .656910
           bits 13 to 20  2536.36       .514      .696426
           bits 14 to 21  2474.80      -.356      .360780
           bits 15 to 22  2468.03      -.452      .325607
           bits 16 to 23  2469.39      -.433      .332570
           bits 17 to 24  2621.85      1.723      .957570
           bits 18 to 25  2361.68     -1.956      .025222
           bits 19 to 26  2501.26       .018      .507093
```

```
     bits 20 to 27  2389.21     -1.567      .058582
     bits 21 to 28  2531.35       .443      .671245
     bits 22 to 29  2613.65      1.607      .945997
     bits 23 to 30  2400.17     -1.412      .079006
     bits 24 to 31  2642.26      2.012      .977883
     bits 25 to 32  2495.45      -.064      .474367

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
     ::               THIS IS A PARKING LOT TEST                        ::
     :: In a square of side 100, randomly "park" a car---a circle of  ::
     :: radius 1.   Then try to park a 2nd, a 3rd, and so on, each     ::
     :: time parking "by ear".  That is, if an attempt to park a car  ::
     :: causes a crash with one already parked, try again at a new    ::
     :: random location. (To avoid path problems, consider parking    ::
     :: helicopters rather than cars.)   Each attempt leads to either ::
     :: a crash or a success, the latter followed by an increment to  ::
     :: the list of cars already parked. If we plot n:  the number of ::
     :: attempts, versus k::  the number successfully parked, we get a::
     :: curve that should be similar to those provided by a perfect   ::
     :: random number generator.  Theory for the behavior of such a   ::
     :: random curve seems beyond reach, and as graphics displays are ::
     :: not available for this battery of tests, a simple characteriz ::
     :: ation of the random experiment is used: k, the number of cars ::
     :: successfully parked after n=12,000 attempts. Simulation shows ::
     :: that k should average 3523 with sigma 21.9 and is very close  ::
     :: to normally distributed.  Thus (k-3523)/21.9 should be a st-  ::
     :: andard normal variable, which, converted to a uniform varia-  ::
     :: ble, provides input to a KSTEST based on a sample of 10.      ::
     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
           CDPARK: result of ten tests on file BLOCKX.RNG
            Of 12,000 tries, the average no. of successes
                 should be 3523 with sigma=21.9
            Successes: 3527    z-score:   .183 p-value: .572463
            Successes: 3480    z-score: -1.963 p-value: .024796
            Successes: 3533    z-score:   .457 p-value: .676028
            Successes: 3514    z-score:  -.411 p-value: .340551
            Successes: 3564    z-score:  1.872 p-value: .969407
            Successes: 3547    z-score:  1.096 p-value: .863437
            Successes: 3512    z-score:  -.502 p-value: .307734
            Successes: 3513    z-score:  -.457 p-value: .323972
            Successes: 3539    z-score:   .731 p-value: .767486
            Successes: 3500    z-score: -1.050 p-value: .146807

           square size   avg. no.  parked   sample sigma
             100.            3522.900       23.039
            KSTEST for the above 10: p=  .006349

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
     ::               THE MINIMUM DISTANCE TEST                         ::
     :: It does this 100 times::   choose n=8000 random points in a    ::
     :: square of side 10000.  Find d, the minimum distance between    ::
     :: the (n^2-n)/2 pairs of points.  If the points are truly inde- ::
     :: pendent uniform, then d^2, the square of the minimum distance ::
```

```
:: should be (very close to) exponentially distributed with mean ::
:: .995 .  Thus 1-exp(-d^2/.995) should be uniform on [0,1) and  ::
:: a KSTEST on the resulting 100 values serves as a test of uni- ::
:: formity for random points in the square. Test numbers=0 mod 5 ::
:: are printed but the KSTEST is based on the full set of 100    ::
:: random choices of 8000 points in the 10000x10000 square.      ::
:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
          This is the MINIMUM DISTANCE test
         for random integers in the file BLOCKX.RNG
Sample no.    d^2     avg     equiv uni
      5    2.8780   1.1865     .944561
     10     .0325    .8289     .032175
     15     .4453    .8742     .360819
     20    3.0833   1.0359     .954897
     25     .4605    .9374     .370478
     30     .2520    .8370     .223770
     35    1.3678    .8025     .747075
     40     .3063    .8304     .264948
     45     .9333    .8562     .608578
     50     .5672    .8187     .434512
     55    1.4760    .8036     .773133
     60     .1485    .7916     .138646
     65    3.5344    .7905     .971337
     70     .1984    .7681     .180790
     75    4.1421    .8526     .984438
     80    1.3541    .8643     .743573
     85     .3786    .8570     .316505
     90     .9284    .8501     .606666
     95    1.7096    .8639     .820608
    100    1.1507    .8585     .685424
MINIMUM DISTANCE TEST for BLOCKX.RNG
     Result of KS test on 20 transformed mindist^2's:
                                p-value= .887840

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
::               THE 3DSPHERES TEST                              ::
:: Choose  4000 random points in a cube of edge 1000.  At each   ::
:: point, center a sphere large enough to reach the next closest ::
:: point. Then the volume of the smallest such sphere is (very   ::
:: close to) exponentially distributed with mean 120pi/3.  Thus  ::
:: the radius cubed is exponential with mean 30. (The mean is    ::
:: obtained by extensive simulation).  The 3DSPHERES test gener- ::
:: ates 4000 such spheres 20 times.  Each min radius cubed leads ::
:: to a uniform variable by means of 1-exp(-r^3/30.), then a     ::
::  KSTEST is done on the 20 p-values.                           ::
:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
          The 3DSPHERES test for file BLOCKX.RNG
 sample no:  1     r^3=   4.664     p-value= .14398
 sample no:  2     r^3=  21.443     p-value= .51070
 sample no:  3     r^3=  70.671     p-value= .90517
 sample no:  4     r^3=  16.043     p-value= .41420
 sample no:  5     r^3=   8.593     p-value= .24907
 sample no:  6     r^3=  13.257     p-value= .35720
 sample no:  7     r^3=  14.018     p-value= .37329
 sample no:  8     r^3=    .223     p-value= .00741
```

```
 sample no:  9     r^3=  37.868     p-value= .71699
 sample no: 10     r^3=  52.088     p-value= .82382
 sample no: 11     r^3=   4.046     p-value= .12617
 sample no: 12     r^3=  22.994     p-value= .53534
 sample no: 13     r^3=  59.855     p-value= .86401
 sample no: 14     r^3= 184.201     p-value= .99785
 sample no: 15     r^3=  56.673     p-value= .84879
 sample no: 16     r^3=  82.638     p-value= .93637
 sample no: 17     r^3=  15.210     p-value= .39771
 sample no: 18     r^3=  51.078     p-value= .81779
 sample no: 19     r^3=  30.468     p-value= .63781
 sample no: 20     r^3=  39.794     p-value= .73459
  A KS test is applied to those 20 p-values.
---------------------------------------------------------
       3DSPHERES test for file BLOCKX.RNG           p-value= .552581
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
     ::      This is the SQEEZE test                                 ::
     ::  Random integers are floated to get uniforms on [0,1). Start- ::
     ::  ing with k=2^31=2147483647, the test finds j, the number of  ::
     ::  iterations necessary to reduce k to 1, using the reduction   ::
     ::  k=ceiling(k*U), with U provided by floating integers from    ::
     ::  the file being tested.  Such j's are found 100,000 times,    ::
     ::  then counts for the number of times j was <=6,7,...,47,>=48  ::
     ::  are used to provide a chi-square test for cell frequencies.  ::
     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
           RESULTS OF SQUEEZE TEST FOR BLOCKX.RNG
         Table of standardized frequency counts
     ( (obs-exp)/sqrt(exp) )^2
        for j taking values <=6,7,8,...,47,>=48:
    -1.5    -1.2    -1.1      .2      .1     -.3
     -.1     -.1      .6    -1.7      .2    -2.0
     1.3      .8     1.0     -.7      .2    -1.0
     -.4      .9     1.3      .2     -.5      .3
    -2.1     1.9     -.1      .2     -.5     -.6
      .6     -.6    -1.1      .3      .1     -.1
     -.2      .2    -1.2     -.1     3.1      .0
      .8
           Chi-square with 42 degrees of freedom: 42.931
              z-score=   .102  p-value= .568901
______________________________________________________________

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
     ::             The  OVERLAPPING SUMS test                        ::
     :: Integers are floated to get a sequence U(1),U(2),... of uni-  ::
     :: form [0,1) variables.  Then overlapping sums,                 ::
     ::   S(1)=U(1)+...+U(100), S2=U(2)+...+U(101),... are formed.    ::
     :: The S's are virtually normal with a certain covariance mat-   ::
     :: rix.  A linear transformation of the S's converts them to a   ::
     :: sequence of independent standard normals, which are converted ::
     :: to uniform variables for a KSTEST. The  p-values from ten     ::
     :: KSTESTs are given still another KSTEST.                       ::
     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
                Test no.  1      p-value  .341673
```

```
                Test no.  2      p-value  .965883
                Test no.  3      p-value  .301137
                Test no.  4      p-value  .580212
                Test no.  5      p-value  .089571
                Test no.  6      p-value  .602217
                Test no.  7      p-value  .390491
                Test no.  8      p-value  .086171
                Test no.  9      p-value  .956151
                Test no. 10      p-value  .002479
   Results of the OSUM test for BLOCKX.RNG
        KSTEST on the above 10 p-values:  .646815

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

     :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
     ::     This is the RUNS test.  It counts runs up, and runs down, ::
     :: in a sequence of uniform [0,1) variables, obtained by float-  ::
     :: ing the 32-bit integers in the specified file. This example   ::
     :: shows how runs are counted:  .123,.357,.789,.425,.224,.416,.95::
     :: contains an up-run of length 3, a down-run of length 2 and an ::
     :: up-run of (at least) 2, depending on the next values.  The    ::
     :: covariance matrices for the runs-up and runs-down are well    ::
     :: known, leading to chisquare tests for quadratic forms in the  ::
     :: weak inverses of the covariance matrices.  Runs are counted   ::
     :: for sequences of length 10,000.  This is done ten times. Then ::
     :: repeated.                                                     ::
     :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
           The RUNS test for file BLOCKX.RNG
     Up and down runs in a sample of 10000
_________________________________________________
                 Run test for BLOCKX.RNG     :
       runs up; ks test for 10 p's: .306738
     runs down; ks test for 10 p's: .740152
                 Run test for BLOCKX.RNG     :
       runs up; ks test for 10 p's: .004499
     runs down; ks test for 10 p's: .155079

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

     :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
     :: This is the CRAPS TEST. It plays 200,000 games of craps, finds::
     :: the number of wins and the number of throws necessary to end  ::
     :: each game.  The number of wins should be (very close to) a    ::
     :: normal with mean 200000p and variance 200000p(1-p), with      ::
     :: p=244/495.  Throws necessary to complete the game can vary    ::
     :: from 1 to infinity, but counts for all>21 are lumped with 21. ::
     :: A chi-square test is made on the no.-of-throws cell counts.   ::
     :: Each 32-bit integer from the test file provides the value for ::
     :: the throw of a die, by floating to [0,1), multiplying by 6    ::
     :: and taking 1 plus the integer part of the result.             ::
     :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
                Results of craps test for BLOCKX.RNG
  No. of wins:  Observed Expected
                                98771    98585.86
                  98771= No. of wins, z-score=  .828 pvalue= .79618
   Analysis of Throws-per-Game:
 Chisq=  17.12 for 20 degrees of freedom, p=  .35474
```

Throws	Observed	Expected	Chisq	Sum
1	66881	66666.7	.689	.689
2	37413	37654.3	1.547	2.236
3	27170	26954.7	1.719	3.955
4	19296	19313.5	.016	3.971
5	13752	13851.4	.714	4.684
6	9843	9943.5	1.017	5.701
7	7128	7145.0	.041	5.741
8	5151	5139.1	.028	5.769
9	3616	3699.9	1.901	7.670
10	2659	2666.3	.020	7.690
11	1929	1923.3	.017	7.707
12	1378	1388.7	.083	7.790
13	1007	1003.7	.011	7.801
14	790	726.1	5.616	13.417
15	539	525.8	.330	13.746
16	394	381.2	.433	14.179
17	294	276.5	1.102	15.282
18	206	200.8	.133	15.415
19	153	146.0	.337	15.752
20	118	106.2	1.308	17.060
21	283	287.1	.059	17.119

SUMMARY FOR BLOCKX.RNG

p-value for no. of wins: .796181

p-value for throws/game: .354742

$$

Results of DIEHARD battery of tests sent to file ZZREPORT.TXT