

duplicate spacings	number observed	number expected	
0	65.	67.668	
1	135.	135.335	
2	129.	135.335	
3	96.	90.224	
4	53.	45.112	
5	15.	18.045	
6 to INF	7.	8.282	

Chisquare with 6 d.o.f. = 2.86 p-value= .174270

For a sample of size 500: mean
 RAND.BIN using bits 3 to 26 2.112

duplicate spacings	number observed	number expected	
0	58.	67.668	
1	134.	135.335	
2	137.	135.335	
3	88.	90.224	
4	50.	45.112	
5	20.	18.045	
6 to INF	13.	8.282	

Chisquare with 6 d.o.f. = 4.90 p-value= .443194

For a sample of size 500: mean
 RAND.BIN using bits 4 to 27 2.058

duplicate spacings	number observed	number expected	
0	72.	67.668	
1	129.	135.335	
2	116.	135.335	
3	103.	90.224	
4	55.	45.112	
5	15.	18.045	
6 to INF	10.	8.282	

Chisquare with 6 d.o.f. = 8.18 p-value= .775019

For a sample of size 500: mean
 RAND.BIN using bits 5 to 28 2.002

duplicate spacings	number observed	number expected	
0	67.	67.668	
1	132.	135.335	
2	137.	135.335	
3	92.	90.224	
4	54.	45.112	
5	8.	18.045	
6 to INF	10.	8.282	

Chisquare with 6 d.o.f. = 7.84 p-value= .750196

For a sample of size 500: mean
 RAND.BIN using bits 6 to 29 2.018

duplicate spacings	number observed	number expected	
0	64.	67.668	
1	140.	135.335	
2	134.	135.335	

3	89.	90.224
4	46.	45.112
5	18.	18.045
6 to INF	9.	8.282

Chisquare with 6 d.o.f. = .47 p-value= .001807
 ::

For a sample of size 500: mean
 RAND.BIN using bits 7 to 30 1.984

duplicate spacings	number observed	number expected
0	69.	67.668
1	142.	135.335
2	131.	135.335
3	83.	90.224
4	48.	45.112
5	18.	18.045
6 to INF	9.	8.282

Chisquare with 6 d.o.f. = 1.32 p-value= .029398
 ::

For a sample of size 500: mean
 RAND.BIN using bits 8 to 31 2.002

duplicate spacings	number observed	number expected
0	69.	67.668
1	133.	135.335
2	127.	135.335
3	100.	90.224
4	48.	45.112
5	18.	18.045
6 to INF	5.	8.282

Chisquare with 6 d.o.f. = 3.12 p-value= .206967
 ::

For a sample of size 500: mean
 RAND.BIN using bits 9 to 32 1.954

duplicate spacings	number observed	number expected
0	60.	67.668
1	138.	135.335
2	149.	135.335
3	96.	90.224
4	37.	45.112
5	15.	18.045
6 to INF	5.	8.282

Chisquare with 6 d.o.f. = 5.94 p-value= .570472
 ::

The 9 p-values were
 .883150 .174270 .443194 .775019 .750196
 .001807 .029398 .206967 .570472
 A KSTEST for the 9 p-values yields .648007

```

::::::::::::::::::::::::::::::::::::::::::::::::::
::          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
:: occurrences 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 RAND.BIN

```

```

For a sample of 1,000,000 consecutive 5-tuples,
chisquare for 99 degrees of freedom=152.737; p-value= .999565

```

```

OPERM5 test for file RAND.BIN

```

```

For a sample of 1,000,000 consecutive 5-tuples,
chisquare for 99 degrees of freedom=198.168; p-value=1.000000

```

```

::
:: 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 RAND.BIN

```

```

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	211	211.4	.000826	.001
29	5299	5134.0	5.302213	5.303
30	23059	23103.0	.083977	5.387
31	11431	11551.5	1.257508	6.645

```

chisquare= 6.645 for 3 d. of f.; p-value= .920839

```

```

::
:: 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 RAND.BIN

```

```

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	197	211.4	.983261	.983
30	5209	5134.0	1.095335	2.079
31	23088	23103.0	.009800	2.088
32	11506	11551.5	.179411	2.268

```

chisquare= 2.268 for 3 d. of f.; p-value= .547328

```

```

:: 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 RAND.BIN

Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG RAND.BIN
b-rank test for bits 1 to 8

	OBSERVED	EXPECTED	(O-E)^2/E	SUM
r<=4	903	944.3	1.806	1.806
r =5	21890	21743.9	.982	2.788
r =6	77207	77311.8	.142	2.930

$p=1-\exp(-SUM/2)= .76894$

Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG RAND.BIN
b-rank test for bits 2 to 9

	OBSERVED	EXPECTED	(O-E)^2/E	SUM
r<=4	945	944.3	.001	.001
r =5	21797	21743.9	.130	.130
r =6	77258	77311.8	.037	.168

$p=1-\exp(-SUM/2)= .08040$

Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG RAND.BIN
b-rank test for bits 3 to 10

	OBSERVED	EXPECTED	(O-E)^2/E	SUM
r<=4	911	944.3	1.174	1.174
r =5	21951	21743.9	1.973	3.147
r =6	77138	77311.8	.391	3.538

$p=1-\exp(-SUM/2)= .82947$

Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG RAND.BIN
b-rank test for bits 4 to 11

	OBSERVED	EXPECTED	(O-E)^2/E	SUM
r<=4	887	944.3	3.477	3.477
r =5	21857	21743.9	.588	4.065
r =6	77256	77311.8	.040	4.106

$p=1-\exp(-SUM/2)= .87163$

Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG RAND.BIN
b-rank test for bits 5 to 12

	OBSERVED	EXPECTED	(O-E)^2/E	SUM
r<=4	908	944.3	1.396	1.396
r =5	21703	21743.9	.077	1.472
r =6	77389	77311.8	.077	1.550

$p=1-\exp(-SUM/2)= .53919$

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

	OBSERVED	EXPECTED	(O-E)^2/E	SUM
r<=4	928	944.3	.281	.281
r =5	21863	21743.9	.652	.934
r =6	77209	77311.8	.137	1.070

$p=1-\exp(-SUM/2)= .41447$

Rank of a 6x8 binary matrix,

rows formed from eight bits of the RNG RAND.BIN

b-rank test for bits 7 to 14

	OBSERVED	EXPECTED	(O-E)^2/E	SUM
r<=4	954	944.3	.100	.100
r =5	21922	21743.9	1.459	1.558
r =6	77124	77311.8	.456	2.015

$p=1-\exp(-\text{SUM}/2)=.63480$

Rank of a 6x8 binary matrix,

rows formed from eight bits of the RNG RAND.BIN

b-rank test for bits 8 to 15

	OBSERVED	EXPECTED	(O-E)^2/E	SUM
r<=4	960	944.3	.261	.261
r =5	21758	21743.9	.009	.270
r =6	77282	77311.8	.011	.282

$p=1-\exp(-\text{SUM}/2)=.13135$

Rank of a 6x8 binary matrix,

rows formed from eight bits of the RNG RAND.BIN

b-rank test for bits 9 to 16

	OBSERVED	EXPECTED	(O-E)^2/E	SUM
r<=4	926	944.3	.355	.355
r =5	21875	21743.9	.790	1.145
r =6	77199	77311.8	.165	1.310

$p=1-\exp(-\text{SUM}/2)=.48048$

Rank of a 6x8 binary matrix,

rows formed from eight bits of the RNG RAND.BIN

b-rank test for bits 10 to 17

	OBSERVED	EXPECTED	(O-E)^2/E	SUM
r<=4	968	944.3	.595	.595
r =5	21846	21743.9	.479	1.074
r =6	77186	77311.8	.205	1.279

$p=1-\exp(-\text{SUM}/2)=.47241$

Rank of a 6x8 binary matrix,

rows formed from eight bits of the RNG RAND.BIN

b-rank test for bits 11 to 18

	OBSERVED	EXPECTED	(O-E)^2/E	SUM
r<=4	940	944.3	.020	.020
r =5	22046	21743.9	4.197	4.217
r =6	77014	77311.8	1.147	5.364

$p=1-\exp(-\text{SUM}/2)=.93157$

Rank of a 6x8 binary matrix,

rows formed from eight bits of the RNG RAND.BIN

b-rank test for bits 12 to 19

	OBSERVED	EXPECTED	(O-E)^2/E	SUM
r<=4	947	944.3	.008	.008
r =5	21434	21743.9	4.417	4.425
r =6	77619	77311.8	1.221	5.645

$p=1-\exp(-\text{SUM}/2)=.94055$

Rank of a 6x8 binary matrix,

rows formed from eight bits of the RNG RAND.BIN

b-rank test for bits 13 to 20

	OBSERVED	EXPECTED	(O-E)^2/E	SUM
r<=4	879	944.3	4.516	4.516
r =5	21404	21743.9	5.313	9.829
r =6	77717	77311.8	2.124	11.953

$p=1-\exp(-\text{SUM}/2)=.99746$

Rank of a 6x8 binary matrix,

rows formed from eight bits of the RNG RAND.BIN

b-rank test for bits 14 to 21

	OBSERVED	EXPECTED	(O-E)^2/E	SUM
r<=4	892	944.3	2.897	2.897
r =5	21757	21743.9	.008	2.905
r =6	77351	77311.8	.020	2.925
p=1-exp(-SUM/2)= .76829				

Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG RAND.BIN

b-rank test for bits 15 to 22

	OBSERVED	EXPECTED	(O-E)^2/E	SUM
r<=4	914	944.3	.972	.972
r =5	21793	21743.9	.111	1.083
r =6	77293	77311.8	.005	1.088
p=1-exp(-SUM/2)= .41951				

Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG RAND.BIN

b-rank test for bits 16 to 23

	OBSERVED	EXPECTED	(O-E)^2/E	SUM
r<=4	986	944.3	1.841	1.841
r =5	21677	21743.9	.206	2.047
r =6	77337	77311.8	.008	2.055
p=1-exp(-SUM/2)= .64217				

Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG RAND.BIN

b-rank test for bits 17 to 24

	OBSERVED	EXPECTED	(O-E)^2/E	SUM
r<=4	954	944.3	.100	.100
r =5	21852	21743.9	.537	.637
r =6	77194	77311.8	.180	.817
p=1-exp(-SUM/2)= .33520				

Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG RAND.BIN

b-rank test for bits 18 to 25

	OBSERVED	EXPECTED	(O-E)^2/E	SUM
r<=4	917	944.3	.789	.789
r =5	21739	21743.9	.001	.790
r =6	77344	77311.8	.013	.804
p=1-exp(-SUM/2)= .33096				

Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG RAND.BIN

b-rank test for bits 19 to 26

	OBSERVED	EXPECTED	(O-E)^2/E	SUM
r<=4	952	944.3	.063	.063
r =5	21769	21743.9	.029	.092
r =6	77279	77311.8	.014	.106
p=1-exp(-SUM/2)= .05146				

Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG RAND.BIN

b-rank test for bits 20 to 27

	OBSERVED	EXPECTED	(O-E)^2/E	SUM
r<=4	972	944.3	.812	.812
r =5	21765	21743.9	.020	.833
r =6	77263	77311.8	.031	.864
p=1-exp(-SUM/2)= .35071				

Rank of a 6x8 binary matrix,
rows formed from eight bits of the RNG RAND.BIN

b-rank test for bits 21 to 28

	OBSERVED	EXPECTED	(O-E)^2/E	SUM
r<=4	961	944.3	.295	.295
r =5	21868	21743.9	.708	1.004
r =6	77171	77311.8	.256	1.260

$p=1-\exp(-SUM/2)= .46741$

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

	OBSERVED	EXPECTED	(O-E)^2/E	SUM
r<=4	941	944.3	.012	.012
r =5	21694	21743.9	.115	.126
r =6	77365	77311.8	.037	.163

$p=1-\exp(-SUM/2)= .07811$

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

	OBSERVED	EXPECTED	(O-E)^2/E	SUM
r<=4	958	944.3	.199	.199
r =5	21703	21743.9	.077	.276
r =6	77339	77311.8	.010	.285

$p=1-\exp(-SUM/2)= .13291$

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

	OBSERVED	EXPECTED	(O-E)^2/E	SUM
r<=4	974	944.3	.934	.934
r =5	21857	21743.9	.588	1.522
r =6	77169	77311.8	.264	1.786

$p=1-\exp(-SUM/2)= .59059$

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

	OBSERVED	EXPECTED	(O-E)^2/E	SUM
r<=4	934	944.3	.112	.112
r =5	21893	21743.9	1.022	1.135
r =6	77173	77311.8	.249	1.384

$p=1-\exp(-SUM/2)= .49942$

TEST SUMMARY, 25 tests on 100,000 random 6x8 matrices
These should be 25 uniform [0,1] random variables:

.768939	.080400	.829466	.871629	.539186
.414466	.634797	.131345	.480485	.472414
.931573	.940547	.997462	.768289	.419511
.642169	.335199	.330964	.051458	.350712
.467412	.078111	.132910	.590595	.499419

brank test summary for RAND.BIN

The KS test for those 25 supposed UNI's yields

KS p-value= .088594

```

:
:
: 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 1:141901 missing words, -.02 sigmas from mean, p-value=
.49224
tst no 2:141829 missing words, -.19 sigmas from mean, p-value= .42556
tst no 3:141470 missing words, -1.03 sigmas from mean, p-value= .15234
tst no 4:141624 missing words, -.67 sigmas from mean, p-value= .25250
tst no 5:142421 missing words, 1.20 sigmas from mean, p-value= .88405
tst no 6:142355 missing words, 1.04 sigmas from mean, p-value= .85113
tst no 7:141697 missing words, -.50 sigmas from mean, p-value= .30991
tst no 8:141394 missing words, -1.20 sigmas from mean, p-value= .11429
tst no 9:142925 missing words, 2.37 sigmas from mean, p-value= .99118
tst no 10:141842 missing words, -.16 sigmas from mean, p-value= .43750
tst no 11:142217 missing words, .72 sigmas from mean, p-value= .76389
tst no 12:141544 missing words, -.85 sigmas from mean, p-value= .19667
tst no 13:141349 missing words, -1.31 sigmas from mean, p-value= .09524
tst no 14:141529 missing words, -.89 sigmas from mean, p-value= .18710
tst no 15:142088 missing words, .42 sigmas from mean, p-value= .66183
tst no 16:141726 missing words, -.43 sigmas from mean, p-value= .33420
tst no 17:142209 missing words, .70 sigmas from mean, p-value= .75809
tst no 18:141675 missing words, -.55 sigmas from mean, p-value= .29202
tst no 19:142298 missing words, .91 sigmas from mean, p-value= .81809
tst no 20:141866 missing words, -.10 sigmas from mean, p-value= .45968

```

(Added, KS Test on the above P values, p=0.337.....)

```

:::
:: 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 QQSO, 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 QQSO
:: by simulation. (Sigma for OPSO, 290, is the true value (to
:: three places), not determined by simulation.
::
::

```

OPSO test for generator RAND.BIN

Output: No. missing words (mw), equiv normal variate (z), p-value (p)

		mw	z	p
OPSO for RAND.BIN	using bits 23 to 32	246403	360.323	1.0000
OPSO for RAND.BIN	using bits 22 to 31	184355	146.364	1.0000
OPSO for RAND.BIN	using bits 21 to 30	148563	22.944	1.0000
OPSO for RAND.BIN	using bits 20 to 29	144510	8.968	1.0000
OPSO for RAND.BIN	using bits 19 to 28	142771	2.971	.9985
OPSO for RAND.BIN	using bits 18 to 27	143070	4.002	1.0000
OPSO for RAND.BIN	using bits 17 to 26	142298	1.340	.9099
OPSO for RAND.BIN	using bits 16 to 25	158870	58.485	1.0000
OPSO for RAND.BIN	using bits 15 to 24	245715	357.951	1.0000
OPSO for RAND.BIN	using bits 14 to 23	183153	142.220	1.0000
OPSO for RAND.BIN	using bits 13 to 22	147801	20.316	1.0000
OPSO for RAND.BIN	using bits 12 to 21	144218	7.961	1.0000
OPSO for RAND.BIN	using bits 11 to 20	142824	3.154	.9992
OPSO for RAND.BIN	using bits 10 to 19	143723	6.254	1.0000
OPSO for RAND.BIN	using bits 9 to 18	142499	2.033	.9790
OPSO for RAND.BIN	using bits 8 to 17	142927	3.509	.9998
OPSO for RAND.BIN	using bits 7 to 16	149694	26.844	1.0000
OPSO for RAND.BIN	using bits 6 to 15	148934	24.223	1.0000
OPSO for RAND.BIN	using bits 5 to 14	143865	6.744	1.0000
OPSO for RAND.BIN	using bits 4 to 13	143352	4.975	1.0000
OPSO for RAND.BIN	using bits 3 to 12	155633	47.323	1.0000
OPSO for RAND.BIN	using bits 2 to 11	167133	86.978	1.0000
OPSO for RAND.BIN	using bits 1 to 10	153209	38.964	1.0000

QQSO test for generator RAND.BIN

Output: No. missing words (mw), equiv normal variate (z), p-value (p)

		mw	z	p
QQSO for RAND.BIN	using bits 28 to 32	141899	-.035	.4860
QQSO for RAND.BIN	using bits 27 to 31	141727	-.618	.2683
QQSO for RAND.BIN	using bits 26 to 30	141454	-1.543	.0614
QQSO for RAND.BIN	using bits 25 to 29	141533	-1.276	.1010
QQSO for RAND.BIN	using bits 24 to 28	141747	-.550	.2911
QQSO for RAND.BIN	using bits 23 to 27	141964	.185	.5735
QQSO for RAND.BIN	using bits 22 to 26	141909	-.001	.4996
QQSO for RAND.BIN	using bits 21 to 25	141625	-.964	.1676
QQSO for RAND.BIN	using bits 20 to 24	142024	.389	.6513
QQSO for RAND.BIN	using bits 19 to 23	142215	1.036	.8499
QQSO for RAND.BIN	using bits 18 to 22	141788	-.411	.3404
QQSO for RAND.BIN	using bits 17 to 21	142142	.789	.7849
QQSO for RAND.BIN	using bits 16 to 20	142046	.463	.6784
QQSO for RAND.BIN	using bits 15 to 19	142042	.450	.6735
QQSO for RAND.BIN	using bits 14 to 18	142123	.724	.7656

QQSO for RAND.BIN	using bits 13 to 17	145956	13.718	1.0000
QQSO for RAND.BIN	using bits 12 to 16	287916	494.938	1.0000
QQSO for RAND.BIN	using bits 11 to 15	289503	500.318	1.0000
QQSO for RAND.BIN	using bits 10 to 14	236612	321.026	1.0000
QQSO for RAND.BIN	using bits 9 to 13	180456	130.667	1.0000
QQSO for RAND.BIN	using bits 8 to 12	145036	10.599	1.0000
QQSO for RAND.BIN	using bits 7 to 11	147387	18.568	1.0000
QQSO for RAND.BIN	using bits 6 to 10	186500	151.155	1.0000
QQSO for RAND.BIN	using bits 5 to 9	420100	943.019	1.0000
QQSO for RAND.BIN	using bits 4 to 8	813826	*****	1.0000
QQSO for RAND.BIN	using bits 3 to 7	834570	*****	1.0000
QQSO for RAND.BIN	using bits 2 to 6	836705	*****	1.0000
QQSO for RAND.BIN	using bits 1 to 5	778706	*****	1.0000

DNA test for generator RAND.BIN

Output: No. missing words (mw), equiv normal variate (z), p-value (p)

		mw	z	p
DNA for RAND.BIN	using bits 31 to 32	142000	.267	.6054
DNA for RAND.BIN	using bits 30 to 31	142231	.949	.8287
DNA for RAND.BIN	using bits 29 to 30	141274	-1.874	.0305
DNA for RAND.BIN	using bits 28 to 29	141577	-.980	.1635
DNA for RAND.BIN	using bits 27 to 28	141741	-.497	.3098
DNA for RAND.BIN	using bits 26 to 27	142194	.840	.7995
DNA for RAND.BIN	using bits 25 to 26	141975	.194	.5768
DNA for RAND.BIN	using bits 24 to 25	142156	.728	.7666
DNA for RAND.BIN	using bits 23 to 24	142818	2.680	.9963
DNA for RAND.BIN	using bits 22 to 23	142193	.837	.7986
DNA for RAND.BIN	using bits 21 to 22	142905	2.937	.9983
DNA for RAND.BIN	using bits 20 to 21	141587	-.951	.1708
DNA for RAND.BIN	using bits 19 to 20	141486	-1.249	.1059
DNA for RAND.BIN	using bits 18 to 19	141385	-1.547	.0610
DNA for RAND.BIN	using bits 17 to 18	141403	-1.494	.0676
DNA for RAND.BIN	using bits 16 to 17	141690	-.647	.2588
DNA for RAND.BIN	using bits 15 to 16	321471	529.680	1.0000
DNA for RAND.BIN	using bits 14 to 15	246401	308.235	1.0000
DNA for RAND.BIN	using bits 13 to 14	216596	220.315	1.0000
DNA for RAND.BIN	using bits 12 to 13	177479	104.925	1.0000
DNA for RAND.BIN	using bits 11 to 12	154197	36.247	1.0000
DNA for RAND.BIN	using bits 10 to 11	146858	14.598	1.0000
DNA for RAND.BIN	using bits 9 to 10	144768	8.433	1.0000
DNA for RAND.BIN	using bits 8 to 9	142419	1.503	.9336
DNA for RAND.BIN	using bits 7 to 8	391301	735.669	1.0000
DNA for RAND.BIN	using bits 6 to 7	612101	*****	1.0000
DNA for RAND.BIN	using bits 5 to 6	669575	*****	1.0000
DNA for RAND.BIN	using bits 4 to 5	661675	*****	1.0000
DNA for RAND.BIN	using bits 3 to 4	578377	*****	1.0000
DNA for RAND.BIN	using bits 2 to 3	510640	*****	1.0000
DNA for RAND.BIN	using bits 1 to 2	478171	991.922	1.0000

```

:
:
: 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 RAND.BIN
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 RAND.BIN      2569.85      .988      .838372
byte stream for RAND.BIN      2454.77      -.640      .261189

```

```

.....
:: 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 3768.86      17.944      1.000000
bits 2 to 9 2644.63      2.045      .979590
bits 3 to 10 2655.62      2.201      .986125
bits 4 to 11 2532.36      .458      .676417
bits 5 to 12 2443.94      -.793      .213939
bits 6 to 13 2524.87      .352      .637464
bits 7 to 14 2572.51      1.026      .847439
bits 8 to 15 2449.01      -.721      .235409
bits 9 to 16 2716.16      3.057      .998882
bits 10 to 17 2506.58      .093      .537081
bits 11 to 18 2667.66      2.371      .991132
bits 12 to 19 2482.48      -.248      .402177
bits 13 to 20 2521.86      .309      .621391
bits 14 to 21 2495.39      -.065      .474030
bits 15 to 22 2478.89      -.299      .382624
bits 16 to 23 2509.04      .128      .550883
bits 17 to 24 2412.05      -1.244      .106777

```



```

:: 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 RAND.BIN

Sample no.	d^2	avg	equiv uni
5	.0073	.7462	.007323
10	.1330	.7885	.125130
15	.2632	.6302	.232453
20	2.7256	.6309	.935387
25	.1534	.5180	.142890
30	.0012	.5970	.001167
35	.1307	.5696	.123107
40	.1100	.5833	.104667
45	2.4114	.6563	.911392
50	.0036	.6986	.003624
55	1.1045	.7952	.670464
60	.2791	.8236	.244603
65	.0002	.8091	.000235
70	.0027	.8382	.002750
75	.0035	.7924	.003506
80	.0032	.8234	.003164
85	.1870	.8423	.171333
90	.0010	.8382	.000975
95	.0002	.9088	.000240
100	2.7592	.9219	.937529

MINIMUM DISTANCE TEST for RAND.BIN

Result of KS test on 20 transformed mindist^2's:
p-value=1.000000

```

:::.....:
::          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 RAND.BIN

sample no: 1	r^3= 2.222	p-value= .07140
sample no: 2	r^3= 73.635	p-value= .91409
sample no: 3	r^3= 3.476	p-value= .10940
sample no: 4	r^3= 37.710	p-value= .71550
sample no: 5	r^3= 6.392	p-value= .19191
sample no: 6	r^3= 57.508	p-value= .85294
sample no: 7	r^3= 4.477	p-value= .13864
sample no: 8	r^3= 23.994	p-value= .55059
sample no: 9	r^3= 18.037	p-value= .45186
sample no: 10	r^3= 39.270	p-value= .72991
sample no: 11	r^3= 32.825	p-value= .66518
sample no: 12	r^3= 15.090	p-value= .39529

```

sample no: 13      r^3= 63.518      p-value= .87964
sample no: 14      r^3= 31.484      p-value= .64988
sample no: 15      r^3= 107.465     p-value= .97218
sample no: 16      r^3= 20.067      p-value= .48772
sample no: 17      r^3= 72.696      p-value= .91136
sample no: 18      r^3= 16.222     p-value= .41769
sample no: 19      r^3= 58.494     p-value= .85770
sample no: 20      r^3= 57.107     p-value= .85096

```

A KS test is applied to those 20 p-values.

```

-----
3DSPHERES test for file RAND.BIN                p-value= .751144
:
: This is the SQUEEZE 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 RAND.BIN
Table of standardized frequency counts
( (obs-exp)/sqrt(exp) )^2
for j taking values <=6,7,8,...,47,>=48:
-1.5  -.7   .3   .6   .5  -1.7
 1.3   .0  2.1   .9  -1.0   .7
  .0   .5   .0  -.2  -.6   -.8
  .3  -.3  -1.0   .1  1.9   .8
 -.7  -.4   .3  -1.8   .8  -.5
 -.8  1.0   .0  -.1  -.6  -.1
  .7  -1.0  -2.0  -1.8  -.6  -1.0
-1.1

```

Chi-square with 42 degrees of freedom: 39.566
z-score= -.266 p-value= .421480

```

:
: 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 .084717
Test no. 2      p-value .054339
Test no. 3      p-value .449485
Test no. 4      p-value .017990
Test no. 5      p-value .660887
Test no. 6      p-value .009312
Test no. 7      p-value .548258
Test no. 8      p-value .472493
Test no. 9      p-value .525066

```

Test no. 10 p-value .436308
 Results of the OSUM test for RAND.BIN
 KSTEST on the above 10 p-values: .965982

```

:
: 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 RAND.BIN
 Up and down runs in a sample of 10000

```

Run test for RAND.BIN :
runs up; ks test for 10 p's: .749861
runs down; ks test for 10 p's: .118684
Run test for RAND.BIN :
runs up; ks test for 10 p's: .845114
runs down; ks test for 10 p's: .717367

```

```

:
: 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 RAND.BIN
 No. of wins: Observed Expected
 98526 98585.86
 98526= No. of wins, z-score= -.268 pvalue= .39446

Analysis of Throws-per-Game:
 Chisq= 19.47 for 20 degrees of freedom, p= .50858

Throws	Observed	Expected	Chisq	Sum
1	66490	66666.7	.468	.468
2	37682	37654.3	.020	.489
3	26846	26954.7	.439	.927
4	19557	19313.5	3.071	3.998
5	13894	13851.4	.131	4.129
6	10006	9943.5	.392	4.521
7	7176	7145.0	.134	4.656
8	5087	5139.1	.528	5.183
9	3619	3699.9	1.767	6.951
10	2722	2666.3	1.164	8.114
11	1926	1923.3	.004	8.118

12	1328	1388.7	2.657	10.775
13	1025	1003.7	.451	11.226
14	688	726.1	2.003	13.229
15	525	525.8	.001	13.231
16	422	381.2	4.378	17.609
17	261	276.5	.873	18.482
18	193	200.8	.305	18.787
19	152	146.0	.248	19.035
20	113	106.2	.433	19.468
21	288	287.1	.003	19.471

SUMMARY FOR RAND.BIN

p-value for no. of wins: .394455

p-value for throws/game: .508585

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