TestU01
1. download testu01.zip from site:http://www.iro.umontreal.ca/~simardr/testu01/tu01.html
2. unzip it, with : unzip testu01.zip
3. go into the new directory, with cd TestU01-1.2.3
4. launch : ./configure
5. launch : make
6. launch : sudo make install
7. You must do the dynamic links of these new libraries, before use it. To do so, launch : sudo ldconfig
8. Create a new file toto.c

```
#include "unif01.h"
#include <stdio.h>
#include <time.h>

Your_PRNG()
{
.
.
.
.
.
.
   return out; //The output of your PRNG,32bit integer(static unsigned int out;)
}




int main()
{
   unif01_Gen *gen;
   gen = unif01_CreateExternGenBits ("prng",Your_PRNG);
//select the following tests
/*
   bbattery_SmallCrush (gen);
   bbattery_Crush (gen);
   bbattery_BigCrush (gen);
   bbattery_Rabbit (gen,1000000000.0);
   bbattery_Alphabit (gen,1000000000.0,0,32);
   bbattery_pseudoDIEHARD(gen);
   bbattery_FIPS_140_2(gen);
*/
   unif01_DeleteExternGenBits (gen);
   return 0;
}
```

8. Put toto.c in theDesktop/TestU01/TestU01-1.2.3/examples/
9. cd Desktop/TestU01/TestU01-1.2.3/examples/
10.gcc toto.c -ltestu01 -lprobdist -lmylib -lm

on the Mesocentre
1.gcc toto.c -o toto -ltestu01 -lprobdist -lmylib -lm
2.cp script_15D.sge toto.sge
3. gedit toto.sge

```
### SGE SCRIPT GENERATOR mesocentre @ univ-fcomte.fr ###

#$ -q normal15d
#$ -l h_rt=360:00:00
#$ -V
#$ -cwd
#$ -o $JOB_NAME.$JOB_ID.out
#$ -e $JOB_NAME.$JOB_ID.err


###################################

### COMMANDES A LANCER, DEBUT ###

###################################

export MY_BINARY=/Data/Users/qwang/toto

$MY_BINARY


###################################

### COMMANDES A LANCER, FIN ###

###################################

exit 0
```

4.qsub toto.sge
5.Find the result in toto.sge.number.out

NIST
1. download sts-2.1.1.zip from
site:http://csrc.nist.gov/groups/ST/toolkit/rng/documentation_software.html
2. unzip it, with : unzip sts-2.1.1.zip
3. go into the new directory, with cd sts-2.1.1
4. launch : make
5.Generate 10^8 bits binary data in file data.txt which consists of bits stored in ASCII format
6. launch : ./assess 1000000

```
        G E N E R A T O R   S E L E C T I O N
        _____________________________________

  [0] Input File              [1] Linear Congruential
  [2] Quadratic Congruential I   [3] Quadratic Congruential II
  [4] Cubic Congruential        [5] XOR
  [6] Modular Exponentiation     [7] Blum-Blum-Shub
  [8] Micali-Schnorr           [9] G Using SHA-1

 Enter Choice: 0


              User Prescribed Input File: data.txt

          S T A T I S T I C A L   T E S T S
          _________________________________

  [01] Frequency                  [02] Block Frequency
  [03] Cumulative Sums              [04] Runs
  [05] Longest Run of Ones          [06] Rank
  [07] Discrete Fourier Transform     [08] Nonperiodic Template Matchings
  [09] Overlapping Template Matchings  [10] Universal Statistical
  [11] Approximate Entropy          [12] Random Excursions
  [13] Random Excursions Variant      [14] Serial
  [15] Linear Complexity

      INSTRUCTIONS
        Enter 0 if you DO NOT want to apply all of the
        statistical tests to each sequence and 1 if you DO.

 Enter Choice: 1

     P a r a m e t e r   A d j u s t m e n t s
     ----------------------------------------
  [1] Block Frequency Test - block length(M):        128
  [2] NonOverlapping Template Test - block length(m): 9
  [3] Overlapping Template Test - block length(m):    9
  [4] Approximate Entropy Test - block length(m):     10
  [5] Serial Test - block length(m):                  16
  [6] Linear Complexity Test - block length(M):       500
```

```
Select Test (0 to continue): 1

Enter Block Frequency Test block length: 20000

    P a r a m e t e r   A d j u s t m e n t s
    ----------------------------------------
 [1] Block Frequency Test - block length(M):         20000
 [2] NonOverlapping Template Test - block length(m): 9
 [3] Overlapping Template Test - block length(m):    9
 [4] Approximate Entropy Test - block length(m):     10
 [5] Serial Test - block length(m):                  16
 [6] Linear Complexity Test - block length(M):       500

Select Test (0 to continue): 0

How many bitstreams? 100

Input File Format:
 [0] ASCII - A sequence of ASCII 0's and 1's
 [1] Binary - Each byte in data file contains 8 bits of data

Select input mode:  0

  Statistical Testing In Progress.....
```

7. When the tests finish, you will find the result in /home/qianxue/Desktop/sts-2.1/experiments/AlgorithmTesting/finalAnalysisReport.txt

Diehard Battery of Tests
1. download Windows Software (732 Kb) diehard.zip from site:http://www.stat.fsu.edu/pub/diehard/
2. unzip it, with : unzip diehard.zip
3. go into the new directory \diehard
4. Your random number generator should produce 32-bit integers in data.txt. You should send them to your ascii file in hex form, 8 hex 'digits' per integer,10 integers per line, no intervening spaces.
5. ASC2BIN.EXE convert data.txt to a binary file data.bin
6. diehard.exe will prompt you to name the file you want to be tested(data.bin) and ask you to select any or all of the 15 tests. You must provide the binary file that DIEHARD expects---a file of 10 to 11 megabytes, that is, a file of at least 80 million bits.