Cryptography Engineering Quiz.5

a) Write a Python/C++ program to generate 1M bytes of cryptographically secure random numbers. -> import secrets

```
cassidy@cassidydeMacBook-Air sts-2.1.2 % /usr/local/bin/python3.12 /Users/cassidy/Desktop/1_密碼工程/Quiz05/RNG.py
```

b) Run the NISTSP800-22 statistical test on your 1M bytes of binary cryptographically secure random numbers and analyze the test results to identify any deviations from the expected statistical properties of random numbers.

1. The progress of running the statistical test

```
cassidy@cassidydeMacBook-Air sts-2.1.2 % ./assess 8388608
         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: random.bin

             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: 65536

     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):         65536
  [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? 1

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:  1

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

   Statistical Testing Complete!!!!!!!!!!!!
```

2. Capture the result (I only took screenshots of some of it.)

```
● cassidy@cassidydeMacBook-Air sts-2.1.2 % cat experiments/AlgorithmTesting/finalAnalysisReport.txt
------------------------------------------------------------------------------
RESULTS FOR THE UNIFORMITY OF P-VALUES AND THE PROPORTION OF PASSING SEQUENCES
------------------------------------------------------------------------------
   generator is <random.bin>
------------------------------------------------------------------------------
 C1  C2  C3  C4  C5  C6  C7  C8  C9 C10  P-VALUE  PROPORTION  STATISTICAL TEST
------------------------------------------------------------------------------
  0   1   0   0   0   0   0   0   0   0    ----      1/1       Frequency
  0   0   0   0   0   0   0   1   0   0    ----      1/1       BlockFrequency
  0   0   0   1   0   0   0   0   0   0    ----      1/1       CumulativeSums
  0   0   1   0   0   0   0   0   0   0    ----      1/1       CumulativeSums
  0   1   0   0   0   0   0   0   0   0    ----      1/1       Runs
  0   0   0   0   0   1   0   0   0   0    ----      1/1       LongestRun
  0   0   0   0   0   0   0   1   0   0    ----      1/1       Rank
  0   0   0   0   0   0   0   0   1   0    ----      1/1       FFT
  0   1   0   0   0   0   0   0   0   0    ----      1/1       NonOverlappingTemplate
  0   0   0   0   1   0   0   0   0   0    ----      1/1       NonOverlappingTemplate
  0   0   0   0   0   0   1   0   0   0    ----      1/1       NonOverlappingTemplate
  0   0   1   0   0   0   0   0   0   0    ----      1/1       NonOverlappingTemplate
  0   0   1   0   0   0   0   0   0   0    ----      1/1       NonOverlappingTemplate
  0   0   0   1   0   0   0   0   0   0    ----      1/1       NonOverlappingTemplate
  0   0   0   1   0   0   0   0   0   0    ----      1/1       NonOverlappingTemplate
  0   0   0   0   0   0   0   1   0   0    ----      1/1       NonOverlappingTemplate
  0   0   0   0   0   1   0   0   0   0    ----      1/1       NonOverlappingTemplate
  0   0   0   0   0   1   0   0   0   0    ----      1/1       NonOverlappingTemplate
  0   0   0   0   0   0   0   1   0   0    ----      1/1       NonOverlappingTemplate
  0   0   0   0   1   0   0   0   0   0    ----      1/1       NonOverlappingTemplate
  0   0   1   0   0   0   0   0   0   0    ----      1/1       NonOverlappingTemplate
  0   0   0   0   0   0   0   1   0   0    ----      1/1       NonOverlappingTemplate
  0   0   0   0   0   0   0   0   1   0    ----      1/1       NonOverlappingTemplate
  0   0   0   0   0   0   0   0   0   1    ----      1/1       NonOverlappingTemplate
  0   0   0   0   0   1   0   0   0   0    ----      1/1       NonOverlappingTemplate
  0   0   0   0   0   0   0   1   0   0    ----      1/1       NonOverlappingTemplate
  0   0   0   0   0   0   0   1   0   0    ----      1/1       NonOverlappingTemplate
  1   0   0   0   0   0   0   0   0   0    ----      1/1       NonOverlappingTemplate
  0   0   0   0   0   0   0   0   1   0    ----      1/1       NonOverlappingTemplate
  0   0   0   1   0   0   0   0   0   0    ----      1/1       NonOverlappingTemplate
  0   0   0   0   1   0   0   0   0   0    ----      1/1       NonOverlappingTemplate
  0   0   0   1   0   0   0   0   0   0    ----      1/1       NonOverlappingTemplate
  0   0   0   0   1   0   0   0   0   0    ----      1/1       NonOverlappingTemplate
  0   0   0   0   0   0   0   0   0   1    ----      1/1       NonOverlappingTemplate
  0   0   0   0   1   0   0   0   0   0    ----      1/1       NonOverlappingTemplate
  0   0   0   1   0   0   0   0   0   0    ----      1/1       NonOverlappingTemplate
  0   0   0   0   0   1   0   0   0   0    ----      1/1       NonOverlappingTemplate
```

```
  0   0   0   0   0   0   0   0   1   0    ----      1/1       NonOverlappingTemplate
  0   0   0   0   0   0   0   0   0   1    ----      1/1       NonOverlappingTemplate
  0   0   0   0   0   0   0   1   0   0    ----      1/1       NonOverlappingTemplate
  0   0   0   0   0   0   0   0   0   1    ----      1/1       NonOverlappingTemplate
  0   0   1   0   0   0   0   0   0   0    ----      1/1       NonOverlappingTemplate
  0   0   0   1   0   0   0   0   0   0    ----      1/1       NonOverlappingTemplate
  0   0   0   1   0   0   0   0   0   0    ----      1/1       NonOverlappingTemplate
  0   0   0   1   0   0   0   0   0   0    ----      1/1       NonOverlappingTemplate
  0   0   0   0   0   0   1   0   0   0    ----      1/1       NonOverlappingTemplate
  0   0   0   0   0   0   0   0   0   1    ----      1/1       NonOverlappingTemplate
  0   0   0   0   0   0   1   0   0   0    ----      1/1       NonOverlappingTemplate
  0   0   0   1   0   0   0   0   0   0    ----      1/1       NonOverlappingTemplate
  0   0   0   0   0   0   1   0   0   0    ----      1/1       NonOverlappingTemplate
  0   0   0   0   0   0   1   0   0   0    ----      1/1       NonOverlappingTemplate
  0   0   0   1   0   0   0   0   0   0    ----      1/1       NonOverlappingTemplate
  0   0   0   0   0   0   0   1   0   0    ----      1/1       NonOverlappingTemplate
  0   0   0   0   0   0   0   0   0   1    ----      1/1       NonOverlappingTemplate
  0   0   0   1   0   0   0   0   0   0    ----      1/1       NonOverlappingTemplate
  0   0   0   0   1   0   0   0   0   0    ----      1/1       NonOverlappingTemplate
  0   0   0   0   1   0   0   0   0   0    ----      1/1       NonOverlappingTemplate
  0   0   0   0   1   0   0   0   0   0    ----      1/1       NonOverlappingTemplate
  0   0   0   0   0   0   1   0   0   0    ----      1/1       NonOverlappingTemplate
  0   0   0   0   0   0   0   0   1   0    ----      1/1       NonOverlappingTemplate
  0   0   0   0   0   0   1   0   0   0    ----      1/1       NonOverlappingTemplate
  0   0   0   0   0   0   0   0   0   1    ----      1/1       NonOverlappingTemplate
  0   0   0   0   0   0   0   1   0   0    ----      1/1       NonOverlappingTemplate
  0   0   1   0   0   0   0   0   0   0    ----      1/1       NonOverlappingTemplate
  0   0   0   0   0   0   0   0   0   1    ----      1/1       NonOverlappingTemplate
  0   0   0   0   1   0   0   0   0   0    ----      1/1       NonOverlappingTemplate
  0   0   0   0   0   1   0   0   0   0    ----      1/1       NonOverlappingTemplate
  0   0   1   0   0   0   0   0   0   0    ----      1/1       NonOverlappingTemplate
  0   0   0   0   0   0   0   1   0   0    ----      1/1       NonOverlappingTemplate
  0   0   0   0   0   1   0   0   0   0    ----      1/1       NonOverlappingTemplate
  0   0   0   0   0   1   0   0   0   0    ----      1/1       NonOverlappingTemplate
  0   0   0   0   0   0   0   0   0   1    ----      1/1       NonOverlappingTemplate
  0   0   1   0   0   0   0   0   0   0    ----      1/1       NonOverlappingTemplate
  0   0   0   0   1   0   0   0   0   0    ----      1/1       NonOverlappingTemplate
  0   0   0   0   1   0   0   0   0   0    ----      1/1       NonOverlappingTemplate
  0   0   1   0   0   0   0   0   0   0    ----      1/1       NonOverlappingTemplate
  0   0   0   0   1   0   0   0   0   0    ----      1/1       NonOverlappingTemplate
  0   0   0   0   0   0   1   0   0   0    ----      1/1       NonOverlappingTemplate
  0   0   0   1   0   0   0   0   0   0    ----      1/1       NonOverlappingTemplate
  0   1   0   0   0   0   0   0   0   0    ----      1/1       NonOverlappingTemplate
  0   0   0   0   1   0   0   0   0   0    ----      1/1       NonOverlappingTemplate
  0   1   0   0   0   0   0   0   0   0    ----      1/1       NonOverlappingTemplate
  0   0   0   0   0   1   0   0   0   0    ----      1/1       NonOverlappingTemplate
```

```
  0   0   0   1   0   0   0   0   0   0    ----      1/1       RandomExcursionsVariant
  0   0   0   0   1   0   0   0   0   0    ----      1/1       RandomExcursionsVariant
  0   0   0   0   0   1   0   0   0   0    ----      1/1       RandomExcursionsVariant
  0   0   0   0   0   0   0   0   0   1    ----      1/1       RandomExcursionsVariant
  0   0   0   0   0   0   0   0   1   0    ----      1/1       RandomExcursionsVariant
  0   0   0   0   0   0   0   1   0   0    ----      1/1       RandomExcursionsVariant
  0   0   0   0   0   0   0   0   0   1    ----      1/1       RandomExcursionsVariant
  0   0   0   0   1   0   0   0   0   0    ----      1/1       RandomExcursionsVariant
  0   0   0   0   1   0   0   0   0   0    ----      1/1       RandomExcursionsVariant
  0   0   0   0   0   0   0   1   0   0    ----      1/1       Serial
  0   0   0   0   0   0   0   1   0   0    ----      1/1       Serial
  0   0   0   0   1   0   0   0   0   0    ----      1/1       LinearComplexity


- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
The minimum pass rate for each statistical test with the exception of the
random excursion (variant) test is approximately = 0 for a
sample size = 1 binary sequences.

The minimum pass rate for the random excursion (variant) test
is approximately = 0 for a sample size = 1 binary sequences.

For further guidelines construct a probability table using the MAPLE program
provided in the addendum section of the documentation.
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

It passed all the static tests, meaning it conforms to every expected statistical property of random numbers. This implies that in the NIST SP800–22 statistical tests, the random number generator passed all tests, demonstrating that the generated random number sequences closely match ideal randomness and perform well across various statistical tests without significant biases or flaws. Therefore, we can confidently assert that the random number sequences generated by this generator possess cryptographic security and can be applied in secure applications requiring high randomness.