"C" library

double X = rand(seed);

(note: AKA random() or uniform())

Returns a random number - let's assume between 0 & 1

Let's be more precise:

- X is a Uniform Random Variable
- A call to rand() produces a random variate of our R.V. X.
  - Producing a Random Variate of a Uniform RV is also referred to as producing a random number.
- A uniform RV is continuous Therefore the range (S_x) of possible values should be infinite

② The 2nd property of an RNG is Variates x_i and x_{i+1} must not be correlated. In other words having knowledge of x_i must not increase the chances of guessing x_{i+1}

· Methods for obtaining RNs.
  - Physical device — white noise level detected by a radio RF receiver
  - Read from a Table of RNs
  - Employ a recursive eq which generates the x_{i+1} From previous x_i
    This is actually deterministic ≡ Pseudo random
    The seed represents the location within the repeatable set of Numbers.

Example Linear Congruent Generators (LCG)

- Let's say we need to generate a number of RN's:

$$x_1, x_2, \cdots, x_N$$

- The time series formed by the RN's is a realization or a sample path of a random process.
  - To make a long story short: each variate, x_i, is actually a random variable.
  - The set of RNs is a family of RV's

- Two crucial properties of a Random Number Generator (RNG)

(1) It must produce a large range (ideally, infinite). Consider if S_X:

$$\{0, 0.2, 0.4, 0.6, 0.8, 1.0\}$$

Laws of probability say, for a continuous RV X

$$P[X = x] = 0$$

In this example, it is discrete such that

$$P[X = x] = \frac{1}{6}$$ (assuming the RNG ensures any possible S_X is possible

Linear Congruent Generators (LCG)

Review: $A \equiv B \pmod{c}$

→ A is congruent to B modulo C

→ A and B are in the same equivalency class

$A = 26 \quad C = 5 \quad B = 11$

$26 \pmod 5 = 1$

$11 \pmod 5 = 1$

$\therefore A \equiv B$

In 1951, Lehmer thought of using the concept to create pseudo random Numbers

$$Z_i = (a Z_{i-1} + c) \pmod m$$

m: modulus
a: multiplier
c: increment

Z_i: remainder
Z_0: Starting value (Seed)

$0 \leq Z_i \leq m-1$ To generate Uniform RNs

$$U_i = Z_i / m$$

Ex. $Z_n = (5Z_{n-1} + 3) \pmod{16}$ $Z_0 = 7$

n	Z_n	U_n
0	7	—
1	6	0.375
2	1	0.063
3	8	0.500
4	11	0.688
5	10	
6	5	
7	12	
8	15	
9	14	
10	9	
11	0	
12	3	
13	2	
14	13	
15	4	0.250
16	7	0.438
17	6	0.375
18	1	0.063
19	8	0.500

$Z_1 = (5Z_0 + 3) \pmod{16}$

$= 6$

$U_1 = \frac{Z_1}{16} = 0.375$

The length of the cycle is the period.

RC4 – uses random permutations during the creation of keystreams.

Review: Permutations are the # of combinations of n items where ordering matters.

For a keysize of 256, there are 256! not repeated permutations.

If the PRNG has full period, this is a very large period! Use Stirling's formula to approx. n!

$$n! \approx n^n e^{-n} \sqrt{2\pi n} \text{ when } n \to \infty$$

$$256! \approx 256^{256} e^{-256} \sqrt{2\pi 256} \approx \underline{10^{500}}$$

(Note: unclear if the it has been proven if this actually is the period of RC4's PRNG)