

**ALGORITHM**

**GENERIC ACTIVITY**



R.NIVETHA

II M.Sc MATHS

20D3238

**GENERATION OF PSEUDO-RANDOM NUMBERS**

**ALGORITHM DEVELOPMENT:**

Random number generators are frequently used in computing science for among other things, testing and analysing the behaviour of algorithms. A sequence of random numbers should exhibit the following behaviour.

1. The sequence should appear as though each number had occurred by chance.
2. Each number should have a specified probability of falling within a given range.

The approach generally taken in computing science for random number generation is to simulate random processes by deterministically producing a sequence of numbers that appear to exhibit random behaviour. These sequences are predictable in advance and for this reason they are usually referred to as pseudo-random sequences. There are many methods for generating pseudo-random numbers. Perhaps the most widely used of these algorithms is the linear congruential method. When this method is appropriately parameterized it will generate pseudo-random sequences that, for practical purposes, satisfy statistical criteria required of uniformly distributed random variables. In a uniform distribution each possible number is equally probable.

The implementation of the linear congruential method is very straight forward. Successive members of the linear congruential sequence {x} are generated using the expression:

= (a+b) mod m for n≥0

Where the parameters a,b,m, and must be carefully chosen in advance according to certain criteria. The parameters a, b, and m are referred to as the multiplier, increment, and modulus respectively.

All parameters should be integers greater than or equal to zero and m should be greater than , a, and b.

**Parameter**

The parameter can be chosen arbitrarily within the range 0 ≤.

**Parameter m**

The value of m should be greater than or equal to the length of the random sequence required. In addition it must be possible to do the computation (a\*x+b) mod m without roundoff.

**Parameter a**

The choice of a depends on the choice of m. If m is a power of 2 then a should satisfy the condition:

a **mod** 8 = 5

If m is a power of 10, then a should be chosen such that:

a **mod** 200 = 21

Further requirements on a are that it should be larger than √m and less than m-√m, (a-1) should be a multiple of every prime dividing into m, and if m is a multiple of 4 then (a-1) should also be a multiple of 4. These conditions, together with the requirement that b should be relatively prime to m are needed to guarantee that the sequence has a period of m.

**Parameter b**

The constant b should be odd and not a multiple of 5.

When a, b, and m are chosen according to the conditions outlined above a sequence of m pseudo-random numbers in the range 0 to (m-1) can be generated before the sequence begins to repeat. A Pascal implementation of the linear congruential method is given below for an m value of 4096.

**Pascal implementation**

**Procedure** random (**var** x: **integer);**

**var**

a {multiplier},

b {increment},

m {modulus}:**integer;**

**begin** {generates pseudo-random numbers x by the linear congruential method}

m := 4096;

{**assert:** 0=<x=<m-1}

b := 853;

a := 109;

x := (a \* x + b) **mod** m

{**assert:** 0=<x=<m-1}

**end**