LAB 2

OBJECTIVE

To generate pseudo-random numbers using Linear Congruential Generator

THEORY

A linear congruential generator (LCG) is an algorithm that yields a sequence of pseudo-randomized numbers calculated with a discontinuous piecewise linear equation. The method represents one of the oldest and best-known pseudorandom number generator algorithms. The theory behind them is relatively easy to understand, and they are easily implemented and fast, especially on computer hardware which can provide modulo arithmetic by storage-bit truncation.

The generator is defined by the recurrence relation :

$$X_{n+1} = (aX_n + c) \mod m$$

Where X is the sequence of pseudorandom values, and

$m,\ 0 < m$ — the "modulus"

$a,\ 0 < a < m$ — the "multiplier"

$c,\ 0 \leq c < m$ — the "increment"

$X_0,\ 0 \leq X_0 < m$ — the "seed" or "start value"

are integer constants that specify the generator. If c = 0, the generator is often called a multiplicative congruential generator (MCG), or Lehmer RNG. If c ≠ 0, the method is called a mixed congruential generator.

CODE

```
// C Program to generate Pseudo-random numbers
// using Linear Congruential Generator.

#include<stdio.h>

void main()
{
    int xi=27;
    int a=17;
    int c=43;
    int m=100;
    int RandNum[m];

    printf("The random numbers are: \n");
    for (int i=0; i<m; i++)
    {
            RandNum[i]= (a*xi+c)%m;
            xi=RandNum[i];
            printf("%d \n",RandNum[i]);
            if (i==0)
                continue;
            if (RandNum[i]==RandNum[0])
            {
                printf("now the pattern repeats..");
                break;
            }
    }
}
```

OUTPUT

```
The random numbers are:
2
77
52
27
2
now the pattern repeats..
```

CONCLUSION

In this lab, I got familiar with pseudo-random numbers and their generation using the Linear Congruential Generator and implemented the algorithm using the C programming language.