Algorithm: random number using Linear Congruential Generator (LCG)

Include predefined class libraries / header files in the program using preprocessor directive #include <iostream> , <cmath>

using namespace std;

Class declaration: declare a class “random”

Private member declaration:

Data variables:

declare long type data variables “i,x,n,a,c,M,y”
declare integer type data variable “seed”

Public member declaration:
Declare member function calculate of type void

Member function definition:
Define member functions calculate using scope resolution operator
“void random::calculate()”

```
{
cout<<"enter the no of random nos. to be generated"<<endl;  //Get user Input “n”
cin>>n;                                                     //Write user input into allocated memory space
cout<<"enter seed";                                         //Get user input “seed”
cin>>seed;
cout<<"random nos are"<<endl;                               // calculations
M=256;                                                      //variable initialization
a=65;
c=27;
x=1;
```

Perform calculations using Least Congruential Generator (LCG)

```
for(i=0;i<=n;i++)                                           //for loop
{
y=((a*seed)+c)%M;
cout<<y<<endl;                                              //Display output
seed=y;
}
}
```

Inside the main function of type integer , create an object of class random and named r. Use this object to call public member functions and terminate the program using return 0 statement

```
int main()
{
random r;          //create object
r.calculate();     //member function calling
return 0;          //program termination
}
```