Practical – 6

**Aim:** Implement a menu driven program to generate random numbers using: a) Triangular distribution b) Uniform distribution.

**Program:**

#include <random>

#include <iostream>

#include <iomanip>

#include <array>

#include <map>

using namespace std;

piecewise\_linear\_distribution<double> triangular\_distribution(double min, double peak, double max)

{

array<double, 3> i{min, peak, max};

array<double, 3> w{0, 1, 0};

return piecewise\_linear\_distribution<double>{i.begin(), i.end(), w.begin()};

}

int main() {

int choice;

cout<<"Enter \n1. For Triangular distribution and \n2. For uniform distribution\n";

cin>>choice;

if(choice&1){

random\_device rd;

mt19937 gen(rd());

auto dist = triangular\_distribution(0, 7, 10);

map<int, int> hist;

for (int i = 0; i < 4000; ++i) {

double num = dist(gen);

++hist[num];

}

cout<<"Following are the random numbers generated : ";

for(auto p : hist) {

cout << p.second/10<< " ";

}

cout<<"\n enter 1 for the graph : \n";

int x;

cin>>x;

if(x&1){

for(auto p : hist) {

cout << setw(2) << setfill('0') << p.first << ' '

<< string(p.second/10,'\*') << '\n';

}

}

}

else{

const int nrolls=500;

const int nstars=95;

const int nintervals=10;

default\_random\_engine generator;

uniform\_real\_distribution<double> distribution(0.0,1.0);

int p[nintervals]={};

cout<<"Random number generated : ";

for (int i=0; i<nrolls; ++i) {

double number = distribution(generator);

++p[int(nintervals\*number)];

cout<<number<<" ";

}

cout<<endl;

cout << "uniform\_real\_distribution (0.0,1.0):" << endl;

cout << fixed; cout.precision(1);

for (int i=0; i<nintervals; ++i) {

cout << float(i)/nintervals << "-" << float(i+1)/nintervals << ": ";

cout << string(p[i]\*nstars/nrolls,'\*') << endl;

}

}

}

**Output:**

