**Samriddhi College**

**Department of Computer Science(BSc.CSIT)**

Lokanthali, Bhaktapur

**LAB REPORT**

**Subject:- Simulation And Modelling**

Submitted By: Submitted To:

TU Roll no:

**Lab-01**

**C program for Random Number generator.**

include<stdio.h>

int main()

{

int a,c,m,x0,i;

printf("Random Number\n\n");

printf("Enter values of a,c,m and x0::\n");

scanf("%d%d%d%d",&a,&c,&m,&x0);

printf("Random numbers are::\n");

for(i=1;i<=4;i++)

{

printf("%d\t",x0);

x0=(a\*x0+c)%m;

}

}

**C Program for Chi-square Test**

#include <iostream>

#include <conio.h>

#include <math.h>

#include <stdlib.h>

using namespace std;

class kstest{

int O[10],E[10],N;

float diff[10];

float chisquare,chitab;

public:

void getdata(int n){

int temp,i;

for(i=0;i<n;i++){

cout<<"Enter frequency of "<<i<<"th value:";

cin>>O[i];

}

N=0;

for(i=0;i<n;i++){

N+=O[i];

}

temp=N/n;

for(i=0;i<n;i++){

E[i]=temp;

}

}

void calculatechi(int n){

int i;

cout<<"\nCalculated differences:";

for(i=0;i<n;i++){

diff[i]=(pow((O[i]-E[i]),2))/E[i];

cout<<"\n"<<diff[i];

}

chisquare=0;

for(i=0;i<n;i++){

chisquare+=diff[i];

}

}

void decide(float chi){

cout<<"\nObtained chi square value:"<<chisquare;

if(chitab>chisquare){

cout<<"\nAccepted :The given distributions are uniform";

}

else{

cout<<"\nRejected:The given distributions are not uniform";

}

}

};

int main(){

kstest calc;

float n,chitab;

cout<<"Chi Square Test\n\n";

cout<<"Enter the number of classes or values:";

cin>>n;

cout<<"Enter the Tabulated value of chi:";

cin>>chitab;

calc.getdata(n);

calc.calculatechi(n);

calc.decide(chitab);

return 0;

}

**C program for K-S test**

#include <stdio.h>

double dplus(double num[], double d1[], int length, double n) {

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

d1[i] = ((i + 1) / n) - num[i];

}

double d1max = d1[0];

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

if (d1max <= d1[i]) {

d1max = d1[i];

}

}

printf("D+ = %f\n", d1max);

return d1max;

}

double dminus(double num[], double d2[], int length, double n) {

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

d2[i] = (num[i] - (i) / n);

}

double d2max = d2[0];

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

if (d2max <= d2[i]) {

d2max = d2[i];

}

}

printf("D- = %f\n", d2max);

return d2max;

}

void kst(double num[], int length) {

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

for (int j = i + 1; j < length; j++) {

if (num[i] > num[j]) {

double temp;

temp = num[i];

num[i] = num[j];

num[j] = temp;

}

}

}

double d1[length];

double d2[length];

double n = (double) length;

double d1max = dplus(num, d1, length, n);

double d2max = dminus(num, d2, length, n);

double dplus = d1max;

double dminus = d2max;

double d,tab;

if (dplus > dminus) {

d = dplus;

printf("D = %f\n", d);

} else {

d = dminus;

printf("D = %f\n", d);

}

printf("Enter the tabulated value::\t");

scanf("%f",&tab);

if(tab<d)

printf("Null hypothesis is accepted.");

else

printf("ALternative hypothesis is accepted.");

}

int main() {

printf("Kolmogorov Test\n\n");

int n;

double dvalue1;

printf("Enter number of elements to compute for test: \t");

scanf("%d", &n);

double num[n];

double dp, dn;

int i;

printf("Enter the elements::");

for (i = 0; i < n; i++) {

scanf("%lf", &num[i]);

}

kst(num, i);

}

**Program for implementation of Markov Chain**

#include <iostream>

using namespace std;

#define n 3

bool checkMarkov(double m[][n])

{

    for (int i = 0; i <n; i++) {

        // Find sum of current row

        double sum = 0;

        for (int j = 0; j < n; j++)

            sum = sum + m[i][j];

        if (sum != 1)

        return false;

    }

    return true;

}

// Driver Code

int main()

{

    // Matrix to check

    double m[3][3] = { { 0, 0, 1 },

                    { 0.5, 0, 0.5 },

                    { 1, 0, 0 } };

    // calls the function check()

    if (checkMarkov(m))

        cout << " yes ";

    else

        cout << " no ";

}

**C Program for Monte Carlo Method**

#include <stdlib.h>

#include <stdio.h>

#include <math.h>

#include <string.h>

#define SEED 35791246

main(int argc, char\* argv)

{

int niter=0;

double x,y;

int i,count=0; /\* # of points in the 1st quadrant of unit circle \*/

double z;

double pi;

printf("Enter the number of iterations used to estimate pi: ");

scanf("%d",&niter);

/\* initialize random numbers \*/

srand(SEED);

count=0;

for ( i=0; i<niter; i++) {

x = (double)rand()/RAND\_MAX;

y = (double)rand()/RAND\_MAX;

z = x\*x+y\*y;

if (z<=1) count++;

}  
 pi=(double)count/niter\*4;printf("# of trials= %d , estimate of pi is %g \n",niter,pi);

}

**Barber Shop Simulation**

**Problem:- Customes arrives at the barber shop at the rate of 18+- 6 barber serves his customer at the rate of 16+-4 minutes. Simulate system for queue of customer and measure the waiting time for 26 customer.**

**Model:**

GENERATE 18,6

QUEUE Chairs

SEIZE Joe

DEPART Chairs

ADVANCE 16,4

RELEASE Joe

TERMINATE 1

**Mechanic Shop Simulation**

**Problem:- Customes arrives at the mechanics shop at the rate of 300+- 200 mechanic serves his customer at the rate of 200+-50 minutes. Simulate system for queue of customer and measure the waiting time for 1000customer.**

GENERATE 300, 200

QUEUE Mechanic

SEIZE Mechanic

DEPART Mechanic

ADVANCE 200, 50

RELEASE Mechanic

TERMINATE 1

**Telephone system simulation**

Problem:-A simple telephone system has 2 external line calls arrives every 100+-60 seconds. When line is occupied the caller redials after 5+-1 minutes have elapsed call duration is 3+-1 minute. A tabulation of distribution of time is caller take to make a successful call is required. How long will it takes for 200 calls to be completed.

Model:

SETS STORAGE 2

TRANSIT TABLE M1,5,1,20,

GENERATE 1.6667,1

AGAIN GATE SNE Sets, occupied, Fry for a line

ENTER SETS

ADVANCE 3,1

LEAVE SETS

TABULATE Transit

TERMINATE 1

OCCUPIED ADVANCE 5.1

TRANSFER AGAIN