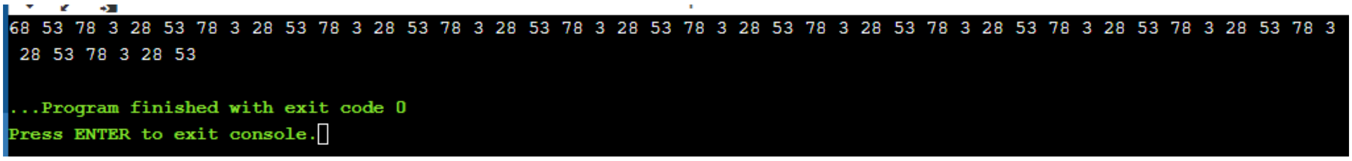
1.WAP to generate 50 random numbers using Mixed Congruential Method where X0=11, m=100, a = 5 and c = 13.

Source code:

```
#include <iostream>

using namespace std;

int main()

{

 int M = 100;

 int a = 5;

 int c=13;

 int X = 11;

 for(int i=0; i<50; i++) {

   X=(a * X + c) % M;

   cout << X << " ";

    }

    return 0;


}
```

OUTPUT:



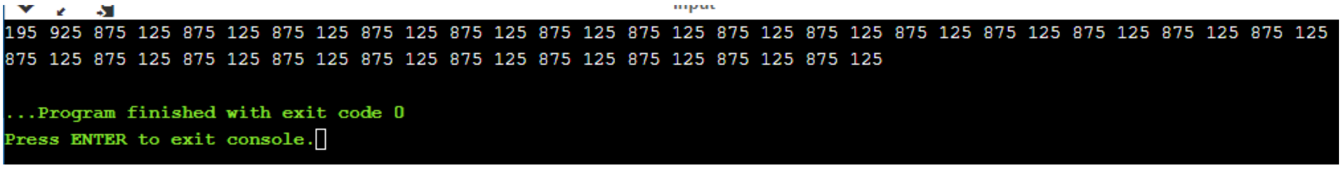
```
68 53 78 3 28 53 78 3 28 53 78 3 28 53 78 3 28 53 78 3 28 53 78 3 28 53 78 3 28 53 78 3 28 53 78 3 28 53 78 3 28 53 78 3
 28 53 78 3 28 53

...Program finished with exit code 0
Press ENTER to exit console.
```

2. WAP to generate 50 random numbers using Multiplicative Congruential Method where X0=13, m =1000, a = 15 and c = 7.

SOURCE SODE:

```
#include <iostream>

using namespace std;

int main()

{

   int M = 1000;

   int a = 15;

   int X = 13;

   int i;

   for(i=0; i<50; i++)    {

      X=(a * X ) % M;

      cout << X << " ";

   }

   return 0;

}
```

Output:



```
195 925 875 125 875 125 875 125 875 125 875 125 875 125 875 125 875 125 875 125 875 125 875 125 875 125 875 125 875 125
875 125 875 125 875 125 875 125 875 125 875 125 875 125 875 125 875 125 875 125

...Program finished with exit code 0
Press ENTER to exit console.
```

3. WAP to implement (i) Kolmogorov – Smirnov test and (ii) Chi-Square Test

i) Kolmogorov – Smirnov test:

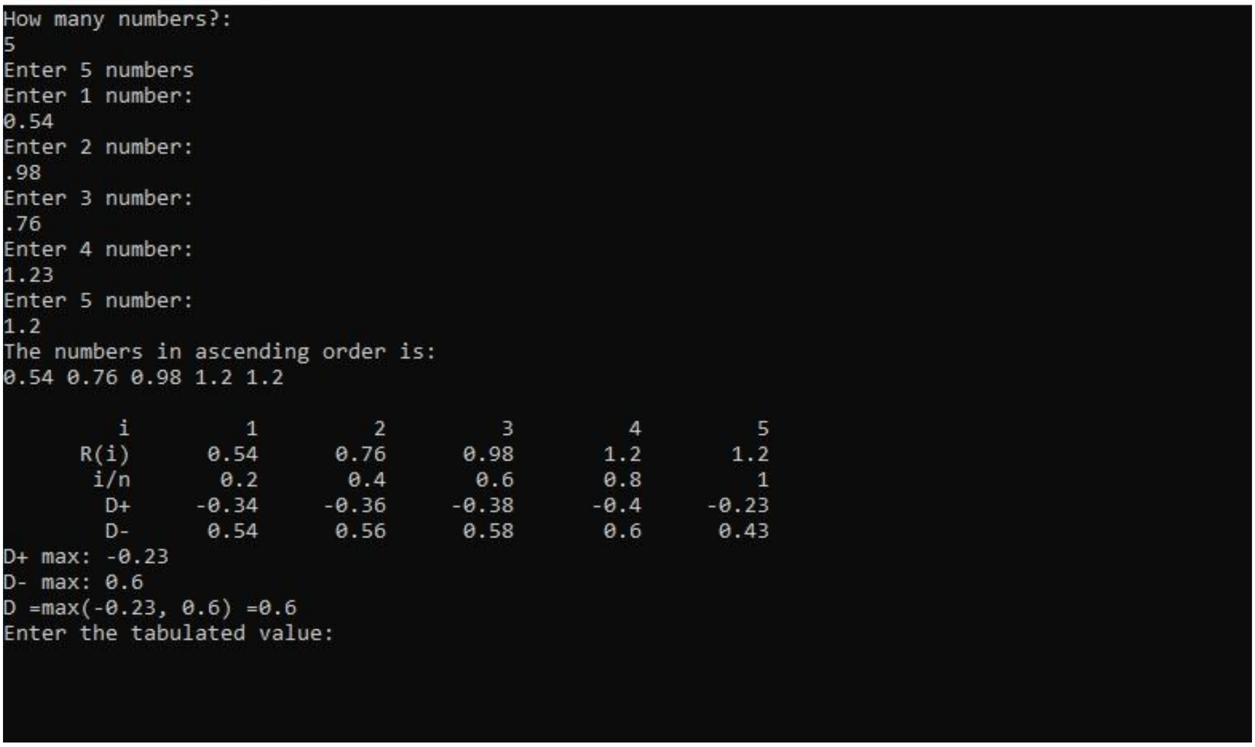
source code:

```
#include<iostream>
#include<conio.h>

#include<iomanip> using

namespace std;

class KS
{
    private:
        float numbers[20];
        float D,tabulatedD; float

        Dplusmax,Dminusmax; float

        Dplus[20],Dminus[20]; float

        ratio[20],ratiominus[20];

        int i,j,n;
    public:
        void getdata() //to get the random numbers
        {
            cout<<"How many numbers?:"<<endl;
            cin>>n;
            cout<<"Enter "<<n<<" numbers"<<endl;

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

            {
                cout<<"Enter "<<i+1<<" number:"<<endl;
                cin>>numbers[i];
            }
        }
        float BubbleSort() // arrange the number in increasing order
        {
  int i,j;    float temp;

            for(i=0;i<n-1;i++)
            {
                for(j=0;j<n-i-1;j++)
                {
                    if(numbers[j]>numbers[j+1])
                    {
```

```
                    temp=numbers[j];

        numbers[j]=numbers[j+1];

                    numbers[j+1]=temp;
                }
            }
        }
        cout<<"The numbers in ascending order is:"<<endl;
        for(i=0;i<n;i++)
        {
            cout<<setprecision(2)<<numbers[i]<<" ";
        }
    }
    void calculate() // find D+, D-
    {
        for(i=0;i<n;i++)
        {
            int j;

        j=i+1;

            ratio[i]=(float)j/n;

ratiominus[i]=(float)i/n;

            Dplus[i]=ratio[i]-numbers[i];
            Dminus[i]=numbers[i]-ratiominus[i];

        }
    }
    void display() // display the tabulated format and find D
    {
        cout<<endl;

cout<<endl;

        cout<<setw(10)<<"i";

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

        {
            cout<<setw(10)<<i;

        }
        cout<<endl; cout<<setw(10)<<"R(i)";

        for(i=0;i<n;i++)
        {
            cout<<setw(10)<<numbers[i];
        }
        cout<<endl;
        cout<<setw(10)<<"i/n";

        for(i=0;i<n;i++)
        {
            cout<<setw(10)<<setprecision(2)<<ratio[i];
        }
        cout<<endl;
        cout<<setw(10)<<"D+";
    for(i=0;i<n;i++)
        {
            cout<<setw(10)<<setprecision(2)<<Dplus[i];
        }
        cout<<endl;
        cout<<setw(10)<<"D-";
for(i=0;i<n;i++)
        {
            cout<<setw(10)<<setprecision(2)<<Dminus[i];
        }
        cout<<endl;
        Dplusmax=Dplus[0];
Dminusmax=Dminus[0];
for(i=1;i<n;i++)
        {
            if(Dplus[i]>Dplusmax)
            {
                Dplusmax=Dplus[i];
            }
            if(Dminus[i]>Dminusmax)
            {
                Dminusmax=Dminus[i];
            }
        }
        cout<<"D+ max: "<<Dplusmax<<endl;
        cout<<"D- max: "<<Dminusmax<<endl;
cout<<"D =max("<<Dplusmax<<", "<<Dminusmax<<") =";
        if(Dplusmax>Dminusmax)
        {
            D=Dplusmax;
        }
        else
```

```
                        {
                                D=Dminusmax;
                        }
                        cout<<D;

                cout<<endl;


                }
        void conclusion() // asking tabulated D and comparing it with D(calculated)
        {
                cout<<"Enter the tabulated value:"<<endl;
                cin>>tabulatedD;

                if(D<tabulatedD)
                {
                        cout<<"The test is accepted."<<endl;
                }
                else
                {
                        cout<<"The test is rejected."<<endl;
                }
        }
};
int main() //main function
{
 KS ks1; //object of KS class  ks1.getdata();

//function calls

        ks1.BubbleSort();

        ks1.calculate();

        ks1.display();

        ks1.conclusion();

        getch();

        return(0);

}
```

Output:



```
How many numbers?:
5
Enter 5 numbers
Enter 1 number:
0.54
Enter 2 number:
.98
Enter 3 number:
.76
Enter 4 number:
1.23
Enter 5 number:
1.2
The numbers in ascending order is:
0.54 0.76 0.98 1.2 1.2

         i         1         2         3         4         5
      R(i)      0.54      0.76      0.98       1.2       1.2
       i/n       0.2       0.4       0.6       0.8         1
        D+     -0.34     -0.36     -0.38      -0.4     -0.23
        D-      0.54      0.56      0.58       0.6      0.43
D+ max: -0.23
D- max: 0.6
D =max(-0.23, 0.6) =0.6
Enter the tabulated value:
```

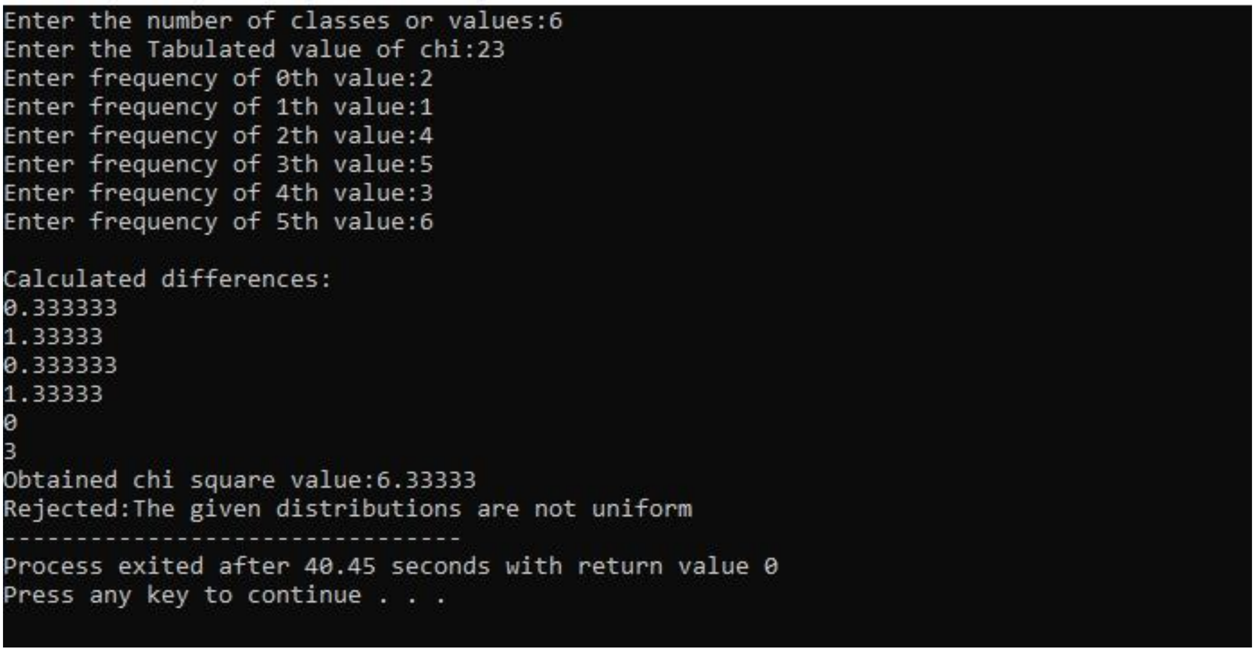
ii) chi-square test:

source code:

```
#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<<"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;
}
```

Output:



```
Enter the number of classes or values:6
Enter the Tabulated value of chi:23
Enter frequency of 0th value:2
Enter frequency of 1th value:1
Enter frequency of 2th value:4
Enter frequency of 3th value:5
Enter frequency of 4th value:3
Enter frequency of 5th value:6

Calculated differences:
0.333333
1.33333
0.333333
1.33333
0
3
Obtained chi square value:6.33333
Rejected:The given distributions are not uniform
--------------------------------
Process exited after 40.45 seconds with return value 0
Press any key to continue . . .
```

4. WAP to implement autocorrelation Test.

SOURCE CODE:

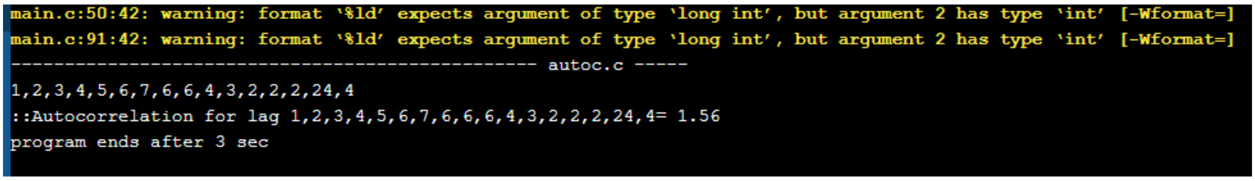
```
#include <stdio.h>
#include <math.h>
#include <stdlib.h
#include <string.h
#define MAX_SIZE
10000000
#define NUM_LAG      100


double    *X;
long int   N;
double     Mean;
double     Variance;


void   load_X_array(void);
double
compute_mean(void);
double
compute_variance(void);
double
compute_autoc(int lag);


void main(void)
{
  double   ac_value;
  int      i;
  X = (double *)
malloc(sizeof(double) *
MAX_SIZE);
  if (X == NULL)
  {
    printf("*** ERROR -
Could not malloc()
enough space \n");

    exit(1);

  }


  load_X_array();

  Mean =
compute_mean();

  Variance =
compute_variance();

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

  {

    ac_value =
compute_autoc(i);

    printf("
Autocorrelation for lag
%4ld = %f \n", i,
ac_value);

  }

  printf("-----------------------
-----------------------------------
------\n");


  free(X);

}


void load_X_array(void)

{

  char
temp_string[1024];


  // Read all values into X
```

```
 N = 0;
 while(1)
 {
   scanf("%s",
temp_string);
   if (feof(stdin)) goto end;
   while
(strcmp(temp_string, "&")
== 0)
   {
     do
     {
       scanf("%s",
temp_string);
       if (feof(stdin)) goto
end;
     } while
(strcmp(temp_string, "&")
!= 0);
     scanf("%s",
temp_string);
     if (feof(stdin)) goto
end;
   }
   X[N] =
atof(temp_string);
   N++;

   if (N >= MAX_SIZE)
   {
     printf("*** ERROR -
greater than %ld data
values \n", MAX_SIZE);
     exit(1);
```

```
    }
  }
  end:
  return;
}
double
compute_mean(void)
{
  double   mean;
  int      i;
  mean = 0.0;
  for (i=0; i<N; i++)
    mean = mean + (X[i] /
N);
  return(mean);
}
double
compute_variance(void)
{
  double   var;
  int      i;
  var = 0.0;
  for (i=0; i<N; i++)
    var = var + (pow((X[i] -
Mean), 2.0) / N);

  return(var);
}

double
compute_autoc(int lag)
```

```
{
  double   autocv;
  double   ac_value;
  int      i;
  autocv = 0.0;
  for (i=0; i<(N - lag); i++)
    autocv = autocv + ((X[i]
- Mean) * (X[i+lag] -
Mean));
  autocv = (1.0 / (N - lag))
* autocv;
  ac_value = autocv /
Variance;


  return(ac_value);
}
```

OUTPUT:



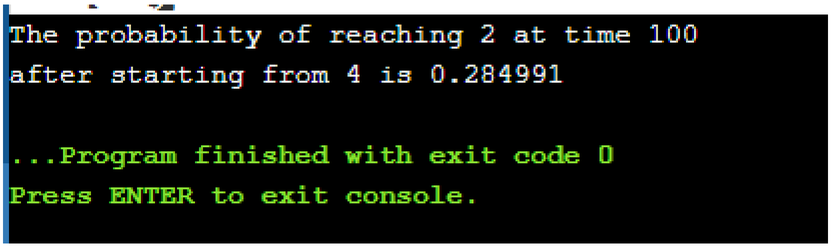
```
main.c:50:42: warning: format '%ld' expects argument of type 'long int', but argument 2 has type 'int' [-Wformat=]
main.c:91:42: warning: format '%ld' expects argument of type 'long int', but argument 2 has type 'int' [-Wformat=]
------------------------------------------------- autoc.c -----
1,2,3,4,5,6,7,6,6,4,3,2,2,2,24,4
::Autocorrelation for lag 1,2,3,4,5,6,7,6,6,6,4,3,2,2,2,24,4= 1.56
program ends after 3 sec
```

7.

```
#include <bits/stdc++.h>

using namespace std;

#define vf vector<float>

   vector<vf >
multiply(vector<vf >
A,    vector<vf > B, int N)

{

    vector<vf > C(N, vf(N, 0));
    for (int i = 0; i < N; ++i)
        for (int j = 0; j < N; ++j)
            for (int k = 0; k < N;
++k)
                C[i][j] += A[i][k] *
B[k][j];
    return C;
}
vector<vf >
matrix_power(vector<vf >
M, int p, int n)
{
    vector<vf > A(n, vf(n, 0));
    for (int i = 0; i < n; ++i)
        A[i][i] = 1;

    while (p) {
        if (p % 2)
            A = multiply(A, M, n);
        M = multiply(M, M, n);
        p /= 2;
    }

    return A;
}
```

```
float
findProbability(vector<vf >
M, int N, int F,

                                    int S,
int T)

{

   vector<vf > MT =
matrix_power(M, T, N);



   return MT[F - 1][S - 1];

}

int main()

{

   vector<vf > G{ { 0, 0.09, 0,
0, 0, 0 },

                 { 0.23, 0, 0, 0, 0,
0.62 },

                 { 0, 0.06, 0, 0, 0, 0
},

                 { 0.77, 0, 0.63, 0,
0, 0 },

                 { 0, 0, 0, 0.65, 0,
0.38 },

                 { 0, 0.85, 0.37,
0.35, 1.0, 0 }};



   int N = 6;
   int S = 4, F = 2, T = 100;



   cout << "The probability of
reaching " << F << " at time
"

             << T << "\nafter
starting from " << S << " is "

             <<
findProbability(G, N, F, S, T);



   return 0;
}
```

OUTPUT:



```
The probability of reaching 2 at time 100
after starting from 4 is 0.284991

...Program finished with exit code 0
Press ENTER to exit console.
```

8. Write a program to calculate measures of a M/M/1 Queue for a given value of Arrival Rate and Service Rate.

SOURCE CODE:

```
#include "cpp.h"

#define NARS 5000
#define IAR_TM 2.0
#define SRV_TM 1.0

event done("done");
facility f("facility");
table tbl("resp tms");
qhistogram  qtbl("num  in
sys", 10l);
int cnt;

void customer();
```

```
extern "C" void sim(int,
char **);

void sim(int argc, char
*argv[])
{
set_model_name("M/M/
1 Queue");
create("sim");
cnt = NARS;
for(int i = 1; i <= NARS; i++)
{
hold(expntl(IAR_TM));
customer();
}
done.wait();
theory();
mdlstat();
}

void customer()
{
float t1;

create("cust");
t1 = clock;
qtbl.note_entry();
f.reserve();
hold(expntl(SRV_TM));
f.release();
tbl.record(clock - t1);
qtbl.note_exit();
if(--cnt == 0)
done.set();
}
```

9. Write a GPSS model to simulate a barber shop where each costumer enters the Shop every 10 ± 2 minutes and a barber takes 13 ± 2 for a haircut. Run the simulation for 1 hour and prepare the report.

BLOCK DIAGRAM:

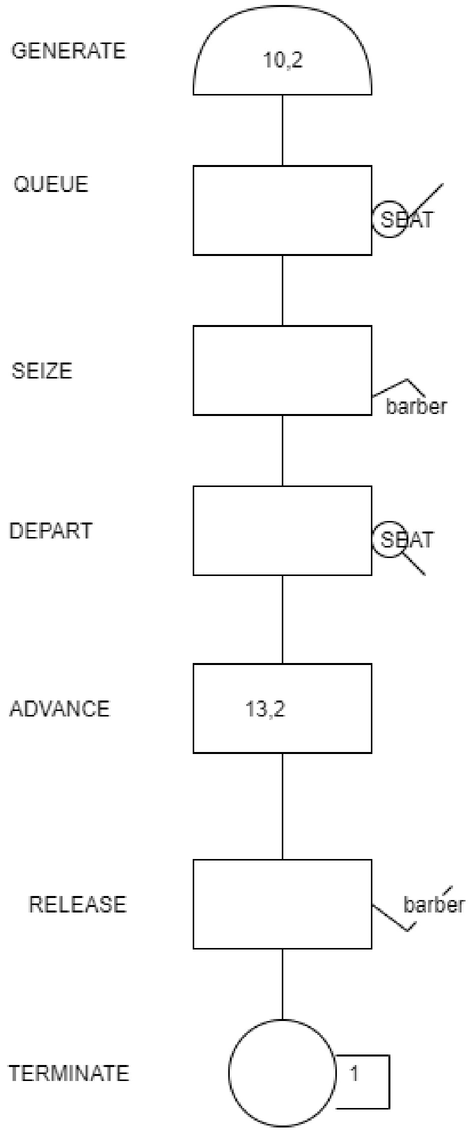
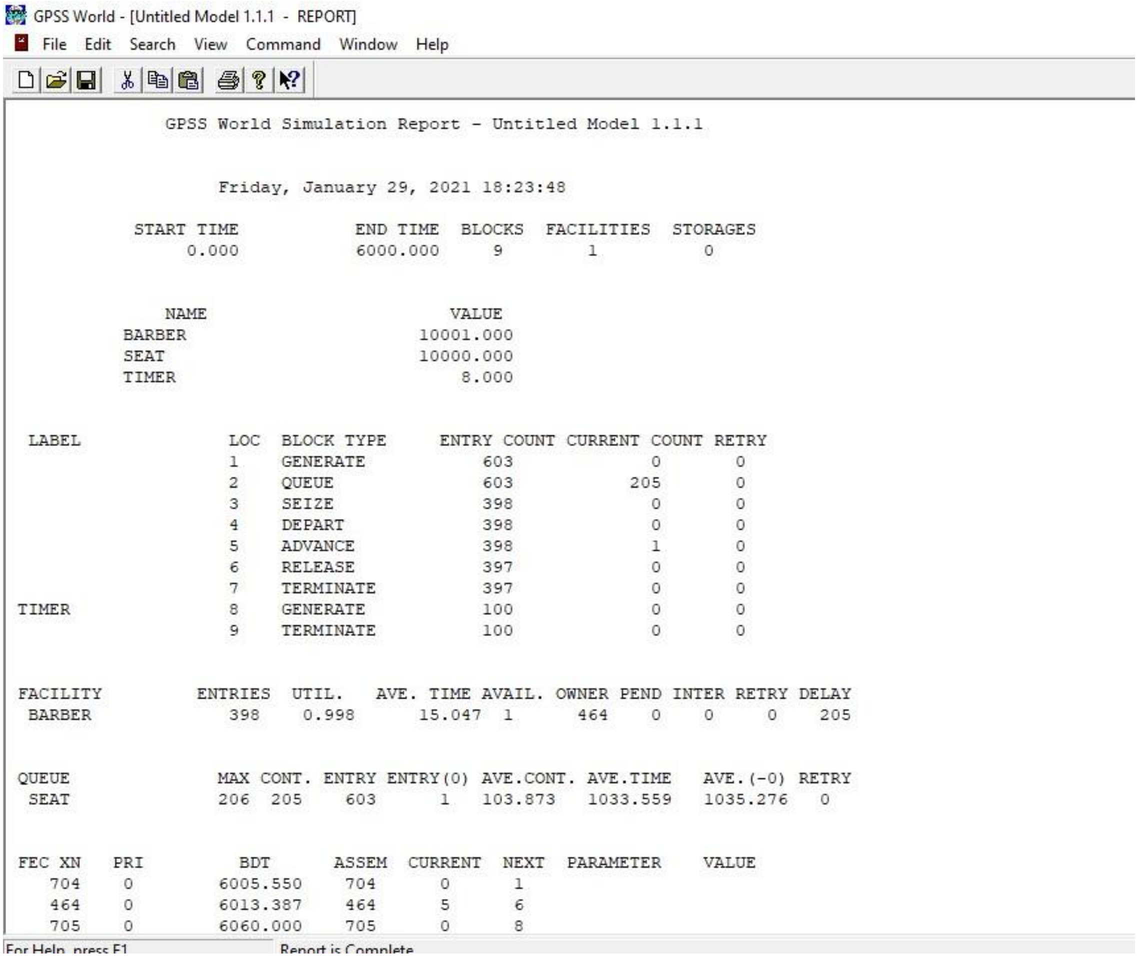


Fig: block diagram of given problem

Code:

```
GENERATE  10,2
 QUEUE    SEAT
SEIZE     BARBER
```

```
DEPART    SEAT
 ADVANCE   15,3
 RELEASE   BARBER
 TERMINATE
 TIMER GENERATE 60
 TERMINATE 1
```


GPSS World - [Untitled Model 1.1.1 - REPORT]
File Edit Search View Command Window Help
For Help, press F1
Report is Complete

```
               GPSS World Simulation Report - Untitled Model 1.1.1


                    Friday, January 29, 2021 18:23:48

            START TIME           END TIME  BLOCKS  FACILITIES  STORAGES
                 0.000           6000.000     9         1          0


               NAME                       VALUE
           BARBER                      10001.000
           SEAT                        10000.000
           TIMER                           8.000


 LABEL              LOC  BLOCK TYPE     ENTRY COUNT CURRENT COUNT RETRY
                    1    GENERATE           603             0       0
                    2    QUEUE              603           205       0
                    3    SEIZE              398             0       0
                    4    DEPART             398             0       0
                    5    ADVANCE            398             1       0
                    6    RELEASE            397             0       0
                    7    TERMINATE          397             0       0
TIMER               8    GENERATE           100             0       0
                    9    TERMINATE          100             0       0


FACILITY         ENTRIES  UTIL.   AVE. TIME AVAIL. OWNER PEND INTER RETRY DELAY
 BARBER            398    0.998      15.047  1      464    0    0     0    205


QUEUE              MAX CONT. ENTRY ENTRY(0) AVE.CONT. AVE.TIME   AVE.(-0) RETRY
 SEAT              206  205    603       1  103.873   1033.559   1035.276   0


FEC XN   PRI         BDT      ASSEM  CURRENT  NEXT  PARAMETER    VALUE
   704    0       6005.550    704      0       1
   464    0       6013.387    464      5       6
   705    0       6060.000    705      0       8
```

10. . Consider a machine tool in a manufacturing shop is turning out parts at the rate of one every 10 minutes. As they are finished, the parts go to an inspector, who takes 7 ± 3 minutes to examine each one and rejects about 10% of the parts. Develop a block diagram and write the code for simulating the above problem using GPSS, and also explain the function of each block used in the block diagram in detail.

Solution:

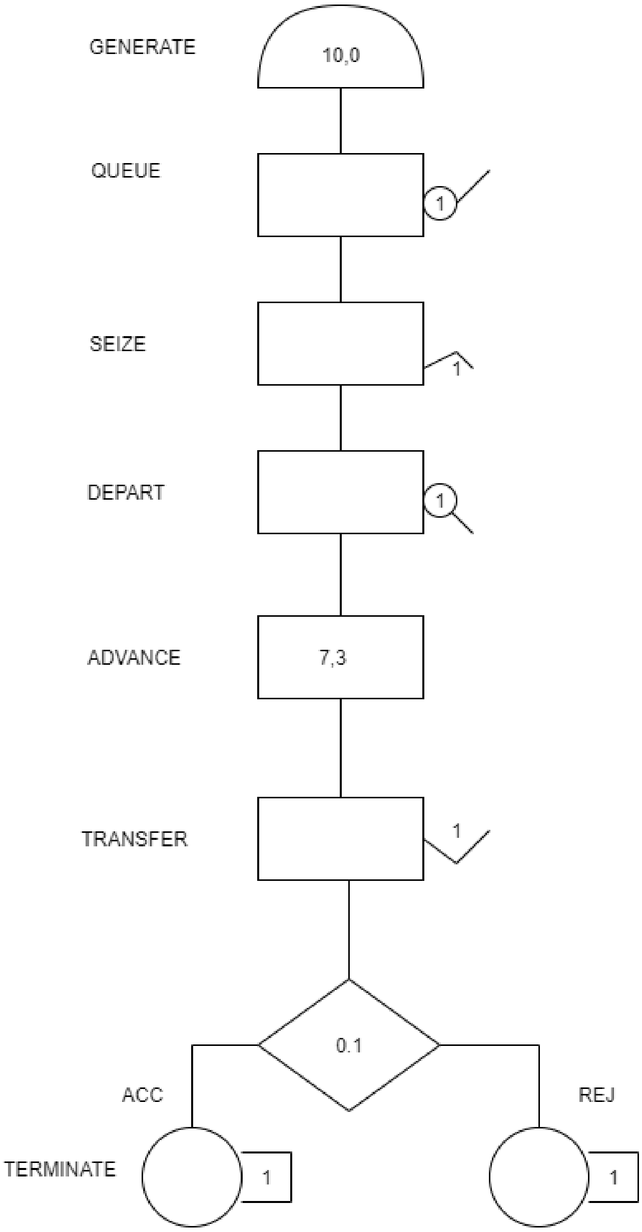
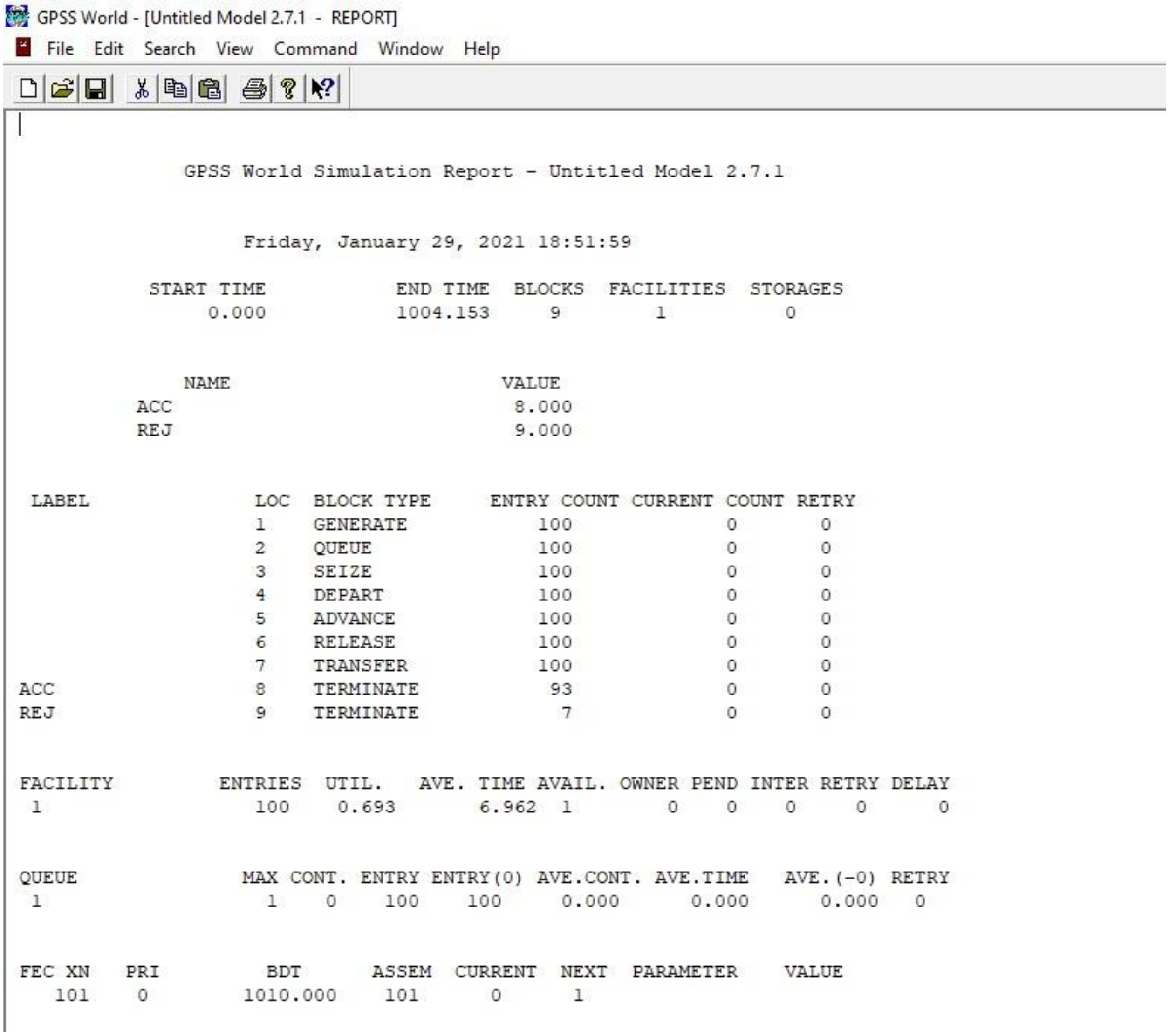


Fig: block diagram for given problem using GPSS

Code:

```
GENERATE    10,0
QUEUE       1
SEIZE       1
DEPART     1
ADVANCE     7,3
```

```
 RELEASE   1
 TRANSFER  0.1 ACC REJ
ACC TERMINATE     1

REJ TERMINATE     1
```



```
              GPSS World Simulation Report - Untitled Model 2.7.1


                   Friday, January 29, 2021 18:51:59

           START TIME           END TIME  BLOCKS  FACILITIES  STORAGES
                0.000           1004.153     9         1          0


              NAME                       VALUE
          ACC                             8.000
          REJ                             9.000


 LABEL              LOC  BLOCK TYPE     ENTRY COUNT CURRENT COUNT RETRY
                    1    GENERATE           100             0       0
                    2    QUEUE              100             0       0
                    3    SEIZE              100             0       0
                    4    DEPART             100             0       0
                    5    ADVANCE            100             0       0
                    6    RELEASE            100             0       0
                    7    TRANSFER           100             0       0
ACC                 8    TERMINATE           93             0       0
REJ                 9    TERMINATE            7             0       0


FACILITY         ENTRIES  UTIL.   AVE. TIME AVAIL. OWNER PEND INTER RETRY DELAY
 1                  100    0.693      6.962  1        0    0     0     0     0


QUEUE              MAX CONT. ENTRY ENTRY(0) AVE.CONT. AVE.TIME   AVE.(-0) RETRY
 1                   1    0    100    100     0.000      0.000      0.000   0


FEC XN   PRI         BDT      ASSEM  CURRENT  NEXT  PARAMETER    VALUE
   101    0       1010.000    101      0       1
```