Lab-8

Aim: Implement a Single Server Queuing System.

CODE:

```
#include<iostream>
#include<iomanip>
#include<cmath>
using namespace std;
double getRandom() {
   return (double(rand())/RAND_MAX);
}
class ExponentialDistribution {
   double mu;
public:
   ExponentialDistribution(double m) {mu=m;}
   double generateRandomVariate() {
      return (-1/mu)*log(getRandom());
   }
};
int main() {
   double meanArrival, meanService, nextArrivalTime=0, totalIdleTime=0, idleTime,
totalWaitTime=0, waitTime, nextDepartureTime=0, nextServiceBeginTime, service,
totalMinutes;
   int requestsServed=0;
   cout<<"Enter Mean Arrival Rate (per hour): "; cin>>meanArrival;
   cout<<"Enter Mean Service Rate (per hour): "; cin>>meanService;
   cout<<"Enter Total Simulation Hours: "; cin>>totalMinutes;
   totalMinutes=totalMinutes*60;
   ExponentialDistribution interArrivalTime(meanArrival/60), serviceTime(meanService/60);

cout<<"R.No.\tArrival_Time\tService_Begin\tService_Time\tDeparture\tWait_Time\tIdle_Ti
me"<<endl;
   while(nextDepartureTime<=totalMinutes)
   {
      nextArrivalTime+=interArrivalTime.generateRandomVariate();
      if(nextArrivalTime<=nextDepartureTime)
      {
         nextServiceBeginTime=nextDepartureTime;
         waitTime=nextDepartureTime-nextArrivalTime;
         totalWaitTime+=waitTime;
         idleTime=0;
      }
      else
      {
         nextServiceBeginTime=nextArrivalTime;
         idleTime=nextArrivalTime-nextDepartureTime;
         totalIdleTime+=idleTime;
         waitTime=0;
```

```
        }
        service=serviceTime.generateRandomVariate();
        nextDepartureTime=nextServiceBeginTime+service;
        ++requestsServed;

cout<<setprecision(5)<<requestsServed<<"\t"<<nextArrivalTime<<"\t\t"<<nextServiceBegi
nTime<<"\t\t"<<service<<"\t\t"<<nextDepartureTime<<"\t\t"<<waitTime<<"\t\t"<<idleTime
<<endl;
    }
    cout<<"Average Wait Time: "<<totalWaitTime/requestsServed<<endl;
    cout<<"Idle Time Percentage: "<<totalIdleTime/totalMinutes*100<<endl;
    cout<<"Capacity Utilization: "<<(nextArrivalTime-
totalIdleTime)/nextArrivalTime*100<<endl;
    return 0;
}
```

OUTPUT:

```
"C:\Users\Ankit Goyal\OneDrive\Documents\labs\8th Sem Lab\SSM\singleServer.exe"
Enter Mean Arrival Rate (per hour): 12
Enter Mean Service Rate (per hour): 14
Enter Total Simulation Hours: 2
R.No.   Arrival_Time    Service_Begin   Service_Time    Departure       Wait_Time       Idle_Time
1       33.418          33.418          2.4576          35.876          0               33.418
2       41.635          41.635          0.90976         42.545          0               5.7599
3       44.316          44.316          3.1467          47.463          0               1.7709
4       49.561          49.561          0.47081         50.032          0               2.0982
5       50.536          50.536          1.2524          51.788          0               0.50415
6       59.276          59.276          0.65165         59.928          0               7.488
7       60.985          60.985          2.8562          63.842          0               1.0573
8       66.939          66.939          18.003          84.942          0               3.0976
9       78.901          84.942          4.3258          89.268          6.0408          0
10      88.477          89.268          7.6988          96.967          0.7906          0
11      88.535          96.967          3.4634          100.43          8.4317          0
12      99.175          100.43          23              123.43          1.2554          0
Average Wait Time: 1.3765
Idle Time Percentage: 45.995
Capacity Utilization: 44.347

Process returned 0 (0x0)   execution time : 7.645 s
Press any key to continue.
```