//||||||||||||||||||||||||||||||||||||||||||||||||||||||||||

// Single Server Queue Simulation in C++ ||

// Programing by : Mohammed D. AL Thobiti ||

// ||

//||||||||||||||||||||||||||||||||||||||||||||||||||||||||||

#include "stdafx.h"

#include<iostream>

using namespace std;

int main()

{

////////////////////////////////////////////////////////

//////////// initilization variables //////////////////

int n=0,n1=0; // no. of jobs

int maxLength=0; // max length queue.

float t=0 ; //clock or time

float eventTime=0,totalTime=0,totalBusytime=0,totalArea=0;

char eventType; //event type : Arraival or Departure

float X=0,U=0,R=0,N=0; // Performence mesures

int nCompletion=0; // how many jobs are completed or departure .

int nRejection=0 ; // how many jobs or custemers are rejected .

int nArrival=0;

float PreEventTime=0;

float ratioRtoA=0; // Ratio between rejected job & arrival jobs.

int Eventlist=0;

int i=0; // counter

//////////////////////////////////////////////////////////

////////////Input & Processing Stage ////////////////////

cout<<"Enter the Total Time or Stop Simulation Time : \n";

cin>>totalTime;

cout<<"Enter the No. of Event list : \n";

cin>>Eventlist;

cout<<"Enter the No. of jobs or custemers : \n";

cin>>n1;

cout<<"Enter the maximume no. of custemers : \n";

cin>>maxLength;

while(i<Eventlist) // while the event list is not empty

{

cout<<"Enter the event type A for 'Arraival' or D 'for 'Departure' \n";

cin>>eventType;

cout<<"Enter the event time \n";

cin>>eventTime;

PreEventTime=t; // time advanced algorithm

t=eventTime;

if(n>0) // server status is busy

{

totalBusytime=totalBusytime+(t-PreEventTime);

totalArea=totalArea+(t-PreEventTime)\*n ;

cout<<"busy time "<<totalBusytime<<" Area="<<totalArea<<endl;

}

else { cout<< " the server is Idle \n"; }

if(eventType=='A' || eventType=='a'){

if(n<maxLength)

{ n++;

nArrival++; }

else {

cout<<"Alert ! the job is rejected \n";

nRejection++ ; }

}

else if(eventType=='D' || eventType=='d'){

n--;

nCompletion++;

cout<<"No. of completion=" <<nCompletion<<endl;

}

else { cout<<"Please Re Enter Event Time (A or D ) \n " ;}

i++; //loop end

}

////////////////////////////////////////////////////////////

///////////////////// RESULTS //////////////////////////////

X=nCompletion/totalTime; // throughput

N=totalArea/totalTime; // average length queue

R=N/X; // response time

U=totalBusytime/totalTime; // utilization

ratioRtoA=(nRejection/nArrival)\*100 ;

cout<<"TotalBusytime = " <<totalBusytime<<endl;

cout<<"Throughput = " <<X<<endl;

cout<<"Average length queue = "<<N<<endl;

cout<<"Response Time = " <<R<<endl;

cout<<"Utilization = " <<U<<endl;

cout<<"Ratio between Rejection jobs and Arrival % = " <<ratioRtoA<<endl;

cout<<"Total time = "<<totalTime<<endl;

cout<<"t="<<t<<endl;

cout<<"Total Area="<<totalArea<<endl;

system("PAUSE");

return 0;

}

//||||||||||||||||||||||||||||||||||||||||||||||||||||||||||

// Simulation of Single Server Queue ||

// Programing by : Mohammed D. AL Thobiti ||

// ||

//||||||||||||||||||||||||||||||||||||||||||||||||||||||||||

