RAMAIAH UNIVERSITY OF APPLIED SCIENCES

ASSIGNMENT

Course Code	CSC304A
Course Name	Computer Simulation
Programme	B.Tech
Department	CSE
Faculty	FET

Name of the Student	Shikhar singh
Reg. No	17ETCS002168
Semester/Year	5th/3rd [2017]
Course Leader/s	Santhoshi Kumari

<table>
<tr><th colspan="5">Declaration Sheet</th></tr>
<tr><td>Student Name</td><td colspan="4">Shikhar singh</td></tr>
<tr><td>Reg. No</td><td colspan="4">17ETCS002168</td></tr>
<tr><td>Programme</td><td colspan="2">B.Tech</td><td>Semester/Year</td><td>05/2017</td></tr>
<tr><td>Course Code</td><td colspan="4">CSC304A</td></tr>
<tr><td>Course Title</td><td colspan="4">Computer Simulation</td></tr>
<tr><td>Course Date</td><td></td><td>to</td><td colspan="2"></td></tr>
<tr><td>Course Leader</td><td colspan="4">Santhoshi Kumari</td></tr>
</table>

Declaration

The assignment submitted herewith is a result of my own investigations and that I have conformed to the guidelines against plagiarism as laid out in the Student Handbook. All sections of the text and results, which have been obtained from other sources, are fully referenced. I understand that cheating and plagiarism constitute a breach of University regulations and will be dealt with accordingly.

<table>
<tr><td>Signature of the Student</td><td></td><td>Date</td><td></td></tr>
<tr><td>Submission date stamp
(by Examination & Assessment Section)</td><td colspan="3"></td></tr>
</table>

Signature of the Course Leader and date	Signature of the Reviewer and date

Declaration Sheet ii
Contents iii
List of Tables iv
List of Figures v
Question No. 1 6
A1.1 Introduction to discrete-time and continuous-time simulation: 6
A1.2 Identify and explain advantage and disadvantages of discrete event simulation and continuous event simulation of a system by taking suitable example: 6
A1.3 Stance taken, justification and Conclusion 7
Question No. 2 8
B1.1 Introduction: 8
B1.2 Simulation Table: 8
B1.3 Result and Analysis 10
Question No. 3 11
B2.1 Introduction to problem solving approach: 11
B2.2 Implementation: 11
B2.3 Results and analysis: 14

Table No.	Title of the table	Pg.No.
Table 1	Statistics when it has rained the day before first day.	
Table 2	Statistics when it has not rained the day before first day.	
Table 3	Simulation over a 10-day time period	

Figure No.	Title of the figure	Pg.No.
Figure 1	A single server queuing system	
Figure 2	Source code	
Figure 3	Source code (updating various fields)	
Figure 4	Source code (printing the details in a tabular format.)	
Figure 5	Source code (analyzing and printing the output statistics.)	
Figure 6	simulation of first 35 requests made.	
Figure 7	Simulation of last 20 requests along with output statistics.	

<u>Question No. 1</u>

Solution to Question No. A:

A1.1 Introduction to discrete-time and continuous-time simulation:

Discrete time simulation is an appropriate approach for simulating systems whose state is discrete and changes at particular time point and then remains in that state for some time. For example, a system is the number of customers in a post office, the number of customers is discrete (integer) and the number of customers only changes when someone enters the post office or finishes its business at the counter.

Continuous time simulation is appropriate approach for systems with a continuous state that changes continuously over time. For example, a system is the amount of liquid in a tank and or its temperature. Such a system can be described by differential equations. Continuous simulation is a technique to solve these equations numerically.

A1.2 Identify and explain advantage and disadvantages of discrete event simulation and continuous event simulation of a system by taking suitable example:

Discrete Event Simulation has evolved as a powerful decision making tool after the appearance of fast and inexpensive computing capacity. Discrete event simulation enables the study of systems which are discrete, dynamic and stochastic. Discrete Event Simulation models are dynamic simulation models. Time evolvement plays an important role in the analyzed system. Stochastic simulation models are run several times to generate a distribution of outcomes that can be analyzed. Discrete Event Simulations are typically used to analysis queuing problems, although it fits to many different applications.

Pros:

· New protocols and different operating procedures or new methods can be analyzed in the simulation model. Meanwhile the real system is not disrupted by experiments.

· Phenomena can be speed up or slowed down in the simulation. An entire shift can be analyzed in minutes.

· The simulation model can be used to train students. The students make their decisions in the simulation model to learn and gain experience before operating the real system. When they make mistakes they can learn from this experience while the real system is not disrupted.

Cons:

· Discrete Event Simulation can be applied only if the simulation model can replicate the reality to a sufficient extent.

· Stochastic skills and an adequate level of experience and knowledge is needed for the creation of a simulation model.

· Like numerical mathematical models and traditional spreadsheet analysis, Discrete Event Simulation only provides estimations for the model outcomes.

Continuous Event Simulation is the basic approach to simulate the event history for each micro unit i.e. the timing of different types of events. In general, a random process is assumed to generate the events being considered, with the probability density for experiencing an event at a given point in time depending on a set of explanatory variables. These may include the state of the unit and the time since the last event of a given type.

For continuous time, the only pro is that continuous models are occasionally easier to work with using pen and paper mathematics, as the real data comes in discrete form, computers work in discrete units. Moreover, Continuous time is a special form of Discrete time.

A1.3 Stance taken, justification and Conclusion

Discrete time simulation and Continuous time simulation, both are useful in their respective domains. In the current usage of simulation, discrete time simulation is evolving rapidly as the simulation has become inexpensive, it doesn't effects the real-time systems, huge simulations can be analyzed in minutes, also real data for simulation comes in discrete units and it is much more efficient to perform simulation in computers than simulating using pen and paper, where a risk of mistake in calculation remains constant. Therefore, Discrete event simulation has become the general form of simulation now a days. So it is better to perform all simulations in discrete-time/discrete-event simulations.

Question No. 2

Solution to Question No. B:

B1.1 Introduction:

The given problem statement is the case of Monte Carlo simulation. Monte Carlo simulation is a technique which incorporates mathematical computation to model the probability of different outcomes in a process that cannot easily be predicted due to the intervention of random variables. It is a technique used to understand the impact of risk and uncertainty in prediction and forecasting models. This type of simulation helps one visualize most of the potential outcomes to have a better insight of the risks involved in the decision-making process.

This problem can be approached by Monte Carlo method because it has a frequency distribution which gives a range to different amount of rainfall. As the random number is provided, it will fall between these predefined ranges and hence makes the random event computational. Which means that the city forecast can be simulated and the whole event can be analyzed.

B1.2 Simulation Table:

Statistics for two scenarios are mentioned in the problem statement.

1. if it rained on the previous day. (as shown in table 1.)
2. if it does not rain on the previous day. (as shown in table 2.)

The cumulative frequencies corresponding to each probability is to be found out for both the cases. Now, after finding out the cumulative frequencies for the given probabilities, we can find out the range in which the random numbers fall into each of the given categories. The cumulative probability and the random number range is mentioned in both the table 1 and 2.

Table 1: stats if it rained on previous day

Rained on previous day			
Event	**Probability**	**Cumulative probability**	**Random no. range**
No Rain	0.50	0.50	0 – 50
1 cm rain	0.25	0.75	51 – 75
2 cm rain	0.15	0.90	76 – 90
3 cm rain	0.05	0.95	91 – 95
4 cm rain	0.03	0.98	96 – 98
5 cm rain	0.02	1.00	99 – 100

Table 2 : stats if it does not rain on the previous day.

Not Rained on previous day			
Event	**Probability**	**Cumulative probability**	**Random no. range**
No Rain	0.75	0.75	0 – 75
1 cm rain	0.15	0.90	76 – 90
2 cm rain	0.06	0.96	91 – 96
3 cm rain	0.04	1.00	97 – 100

Now, based upon the two tables mentioned above, the city weather is simulated for 10 days as shown in the table 3. This table consists of 5 columns which shows the number of days, the random digit chosen between 10 to 100, the event (if it rained or not), the amount of rainfall to actually occur and the explanation as to from which table the statistics are being referred.

NOTE: as per the question, it was assumed that it didn't rain on the day before the first day.

The random digits chosen for the simulation are as follows: -
49, 63, 86, 93, 69, 78, 51, 22, 12, 10.

Table 3: simulation table for 10 days.

Simulating the city weather for 10 days				
DAY	**Random digit**	**EVENT (rain / no rain)**	**Amount Of rain (in cm)**	**explanation**
1	49	No rain	0	No rain on the previous day. (assumption)
2	63	No rain	0	No rain on the 1st day (table 2)
3	86	Rain	1	No rain on the 2nd day (table 2)
4	93	Rain	3	Rained on the 3rd day (table 1)
5	69	Rain	1	Rained on the 4th day (table 1)
6	78	Rain	2	Rained on the 5th day (table 1)
7	51	Rain	1	Rained on the 6th day (table 1)
8	22	No rain	0	Rained on the 7th day (table 1)
9	12	No rain	0	No rain on the 8th day (table 2)
10	72	No rain	0	No rain on the 9th day (table 2)

		∑ no. of days without rain = 5 days	∑ amount of rain = 8 cm	

B1.3 Result and Analysis

The random digits chosen for the problem were 49, 63, 86, 93, 69, 78, 51, 22, 12, 10.
The simulation table was made along with the details corresponding to each random digit and at the last, total number of days without rain and total amount of rainfall was calculated.

From the above table 3, the following things can be deduced: -

- Total rain during the time period of 10 days was **8 cms**.
- Total number of days the city lasted without rain in the time period of 10 days was **5 days**.
- Average rainfall each day = $\frac{\sum amount\ of\ rain\ each\ day}{total\ no.\ of\ days}$ = $\frac{8}{10}$ = **0.8 cm.**
- Average time span it did not rain = $\frac{\sum total\ no\ .of\ days\ it\ does\ not\ rain}{total\ no.\ of\ days}$ = $\frac{5}{10}$ = **0.5 day i.e. on an average it didn't rain for half a day**.

The accuracy of Monte Carlo method of simulating random distributions is analyzed.

Although Monte Carlo adds mathematical weightage to the analysis of risk factors involved in a system, it can be seen that the accuracy of some standard computer codes is overstated. It has been shown that increasing the accuracy of the Monte Carlo method by increasing the number of simulations is not economical because the time complexity of algorithm is inversely proportional to the squared relative error[2] . Therefore, a new method of increasing the accuracy with a significant reduction in the number of simulations was developed. For the new method, the number of simulations and time complexity of algorithm are inversely proportional to the relative error.

<u>Question No. 3</u>

<u>Solution to Question No. 3:</u>

B2.1 Introduction to problem solving approach:

An ATM simulation system can be described as discrete single server queueing system. Which means that service requests are going to be at discreet point in time i.e. a state change (event) is going to be unique and discontinuous. Also, only once machine (ATM) is handling all the requests hence it is a single server system. Discrete event simulation is generally carried out by a software designed in high level programming languages such as Pascal, C++, or any specialized simulation language. Following are the five key features −

1. Entities − These are the representation of real elements like the parts of machines.
2. Relationships − It means to link entities together.
3. Simulation Executive − It is responsible for controlling the advance time and executing discrete events.
4. Random Number Generator − It helps to simulate different data coming into the simulation model.
5. Results & Statistics − It validates the model and provides its performance measures.

Simulation of a single server queuing system:

A queue is the combination of all the entities in the system being served and those waiting for their turn. The central element of the system is a server, which provides service to the connected devices or items. Items request to the system to be served, if the server is idle. Then, it is served immediately, else it joins a waiting queue. After the task is completed by the server, the item departs. A simple queuing system can be represented by the following figure:

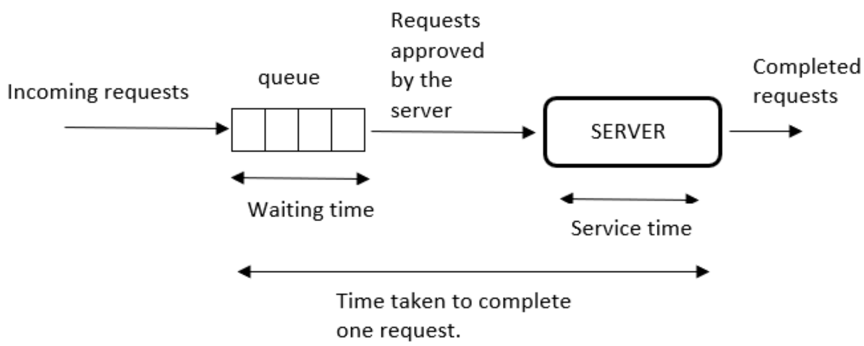

Figure 1: a single server queuing system.

As all the requests will be made randomly by people, hence each person is going to make request one by one at each point of time. Other will have to be in a queue until the person who reaches first, completes the service. Thus, the working of this system is based on FIFO (First in First Out) algorithm.

B2.2 Implementation:

As mentioned in the problem statement, the simulation is to be performed for 12 hours i.e. 720 mins in total. The interarrival time of the incoming customer (service request) is going to be chosen randomly between 1 to 4 mins and service time for each request will be between 1 to 6 min.

ALGORITHM:

1. Record the interarrival time and update the arrival time of the request and send the request to the server for computation.

2. Check whether the server is idle or not.
    a. If the server is idle, start the service and update the start time.
    b. If the server is not idle, push the service in the queue and wait for the ongoing service to end.
3. Add service time to the arrival time to get the end time of service.
4. Note down the difference of arrival time and service end time to get time spent by each request in the system.
5. Note down the waiting time of each service while in queue.
6. Note down the server idle time.
7. Repeat steps 1 to 7 until last request arrives.
8. After the last request, stop the simulation and compute the output stats i.e.
    a. Total Idle time of the ATM machine (server).
    b. Average waiting time of a customer.
    c. Average service time.
    d. Average time between arrivals.
    e. Average time a customer spends in the system.
    f. Percentage utilization of the whole system.

$$\text{Percentage utilization} = \frac{\sum(average\ service\ time)}{\sum(average\ interarrival\ time)} * 100$$

Code implementation:

```java
package single_server_system;

/**
 *
 * @author shikhar
 */
import java.util.Random;
import java.util.Scanner;

public class Single_server_system {

    public static void main(String[] args) {

      Scanner in = new Scanner (System.in);

      Random rand = new Random();

      System.out.printf("Simulating the ATM Machine  ...\n");

      int request=0;
      int total=0;

      int[] iat=new int[10000], st =new int[10000],at= new int[10000],ss= new int[10000];

      int[] se = new int[10000], wait =new int[10000], idle = new int[10000], tcss = new int[10000];

      double sumwait=0.00, sumidle=0.00,sumservice=0.00,sumiat=0.00, sumtcss=0.00;

      se[0]=0;


      System.out.printf("Reqno \tIAT \tST \tAT \tSS \tSE \tTCSS \twait \tidle\n");
```
Figure 2: source code

```java
for(int i=1;se[i-1]<=720;i++)
{

    iat[i] = rand.nextInt(4)+1; //random number generation

    st[i] = rand.nextInt(6)+1; //random number generation

    total += iat[i]; // cummilative sum

    at[i]=total; //initilization

    if((at[i]-se[i-1])>=0)
        ss[i]=at[i]; //check and then initilize start time

    else
        ss[i]=se[i-1];

    se[i]=ss[i]+st[i];  //calculating service end

    wait[i]=ss[i]-at[i]; //calculating wait time

    idle[i]=ss[i]-se[i-1]; //calculating idle time

    tcss[i] = se[i]-at[i]; // calculating the time a customer spends in a system

    sumwait+=wait[i];  // calculating total wait time

    sumidle+=idle[i]; // calculating idle wait time

    sumservice+=st[i]; // calculating total service time

    sumiat+=iat[i]; // calculating total inter arrival time

    sumtcss+=tcss[i]; // calculating total time a customer spends in system
```

Figure 3: source code (updating various fields)

```java
    System.out.printf("%d \t%d \t%d \t%d \t%d \t%d \t%d \t%d \t%d\n",i,iat[i],st[i],at[i],ss[i],se[i],tcss[i],wait[i],idle[i]);

    request=i; // updating flag for the last request.
}
```

Figure 4: source code (printing the details in a tabular format.)

```java
System.out.printf("\nThe total number of requests processed: %d",request);
System.out.printf("\nAverage waiting time of a customer: %f\n",(sumwait/request));
System.out.printf("Idle time of the ATM machine: %f\n",sumidle);
System.out.printf("Average service time: %f\n",(sumservice/request));
System.out.printf("Average time between arrivals: %f\n", sumiat/request);
System.out.printf("Average time customer spends in the system: %f\n", sumtcss/request);
System.out.printf("Percentage utilization of the system: %f  \n", sumiat*100/sumservice);
```

Figure 5: source code (computing and printing output stats.)

All the figures mentioned above are the different functionalities of the program.

B2.3 Results and analysis:

The program was run and successfully executed, which yielded the following output: -

```
run:
Simulating the ATM Machine  ...
Reqno   IAT     ST      AT      SS      SE      TCSS    wait    idle
1       3       3       3       3       6       3       0       3
2       3       1       6       6       7       1       0       0
3       3       3       9       9       12      3       0       2
4       4       6       13      13      19      6       0       1
5       2       5       15      19      24      9       4       0
6       4       1       19      24      25      6       5       0
7       2       3       21      25      28      7       4       0
8       2       4       23      28      32      9       5       0
9       4       5       27      32      37      10      5       0
10      3       4       30      37      41      11      7       0
11      2       2       32      41      43      11      9       0
12      1       2       33      43      45      12      10      0
13      3       1       36      45      46      10      9       0
14      4       3       40      46      49      9       6       0
15      3       1       43      49      50      7       6       0
16      3       6       46      50      56      10      4       0
17      3       4       49      56      60      11      7       0
18      4       4       53      60      64      11      7       0
19      1       5       54      64      69      15      10      0
20      2       5       56      69      74      18      13      0
21      2       5       58      74      79      21      16      0
22      1       1       59      79      80      21      20      0
23      1       2       60      80      82      22      20      0
24      1       2       61      82      84      23      21      0
25      4       6       65      84      90      25      19      0
26      2       2       67      90      92      25      23      0
27      2       3       69      92      95      26      23      0
28      1       1       70      95      96      26      25      0
29      1       1       71      96      97      26      25      0
30      4       2       75      97      99      24      22      0
31      2       6       77      99      105     28      22      0
32      1       2       78      105     107     29      27      0
33      3       1       81      107     108     27      26      0
34      3       5       84      108     113     29      24      0
35      1       5       85      113     118     33      28      0
```

Figure 6: simulation of first 35 requests made.

```
182     1       4       478     636     640     162     158     0
183     2       1       480     640     641     161     160     0
184     4       5       484     641     646     162     157     0
185     2       6       486     646     652     166     160     0
186     4       6       490     652     658     168     162     0
187     3       2       493     658     660     167     165     0
188     3       5       496     660     665     169     164     0
189     1       6       497     665     671     174     168     0
190     4       6       501     671     677     176     170     0
191     2       6       503     677     683     180     174     0
192     3       1       506     683     684     178     177     0
193     4       6       510     684     690     180     174     0
194     1       4       511     690     694     183     179     0
195     3       5       514     694     699     185     180     0
196     3       3       517     699     702     185     182     0
197     4       1       521     702     703     182     181     0
198     3       3       524     703     706     182     179     0
199     4       3       528     706     709     181     178     0
200     2       5       530     709     714     184     179     0
201     2       6       532     714     720     188     182     0

The total number of requests processed: 201
Average waiting time of a customer: 90.820896
Idle time of the ATM machine: 4.000000
Average service time: 3.587065
Average time between arrivals: 2.651741
Average time customer spends in the system: 94.407960
Percentage utilization of the system: 73.925104
BUILD SUCCESSFUL (total time: 0 seconds)
```

Figure 7: simulation of the last 20 requests along with the output stats.

The output shown in the figures above consists of a tabular format where different columns indicate different attributes of the simulation table, which are as follows: -

- Reqno. : request number
- IAT : interarrival time
- ST : Service time
- AT : arrival time
- SS : service start
- SE : service end
- TCSS : time spent by customer in the system
- Wait : waiting time (time spent by the customer in the queue.)
- Idle : idle time (number of minutes the server remained idle as it didn't receive any request.)

The simulation was done till 12 hours i.e. 720 mins and during this time, a total of 201 service requests were successfully completed. All the service requests after 720th min were dismissed. And the customer whose service end time exceeds 720 mins was also dismissed.

Also, the following output statistics were computed: -

- Total number of requests processed : 201
- Average waiting time of a customer : 90.820896 mins
- Idle time of the ATM machine : 4.00 mins
- Average service time : 3.587065 mins
- Average time between arrivals : 2.651741 mins
- Average time a customer spends in the system : 94.407960 mins
- Percentage utilization of the system : 73.9251 %

The single server queuing system is the simplest queuing system in use. It has many advantages like reducing queue flow traffic, providing the users (services, users etc. Here, customers) a more relaxed environment etc. But it has its own disadvantages which sometimes, far outreach the advantages. The price to avail the system will be expensive and the system software has the tendency to crash and be corrupted. Moreover, the cost of maintenance will consume a part of the business profit. Also, one limitation is the possibility that the waiting space may in fact be limited. Another possibility is that arrival rate is state dependent. That is, potential customers are discouraged from entering the queue if they observe a long line at the time they arrive. Another practical limitation of the model is that the arrival process is not stationary. It is quite possible that the service station would experience peak periods and slack periods during which the arrival rate is higher.

Bibliography

---

1. https://en.wikipedia.org/wiki/Monte_Carlo_method#Monte_Carlo_and_random_numbers
2. "Accuracy and efficiency of Monte Carlo method" – Julius Goodman
   https://inis.iaea.org/collection/NCLCollectionStore/_Public/19/047/19047359.pdf
3. https://www.academia.edu/35688542/Queue_Management_Systems_Characteristics_Advantages_and_Application