MMU®

MULTIMEDIA UNIVERSITY

TMA1301 - TT07
Group 4
2019/20

Title:
Queue System for Ticket Purchasing

Mohammad Luqman bin Mohd Salizan	1151102443
Azizi bin Mohamad Amin	1181302515
Muammar Ghadafi bin Badrul Hisham	1151101837
Muhammad Emir Aizzat bin Abdillah	1181302265
Nabil Aizat bin Mohd Hafiz	1171100507

Table of Contents

Details and Discussion

This report details the simulation system for ticket purchasing. It simulates customer's arrival times, waiting times, inter-arrival times and time spent waiting getting the average for all and also calculates the average time of a customer has spent in the system. For each simulation cycle, there are a number of customers that need to be served. Each customer will arrive at any time in the simulation cycle. After a customer arrives, the customer will be served if one of the counters is empty. If none of the counters are empty, the customer needs to wait until the counter is empty. When the customer is served, the customer can have different service time. After the service time ended, the customer left the counter.

Based on the scenario presented, there will be two probability graphs needed for the customer, which are the arrival time and the time taken for the customer to finish the service. The probability of the arrival time is simulated by setting the probability of the inter arrival time between customers. The inter-arrival will set a time between the current customer and the next customer, how far apart in time are they. The time taken for the customer to finish the service is also simulated by having its own probability graph. This will set how long the customer occupied the counter before they left the counter empty. The simulation also allows restrictions to the maximum range of inter-arrival time and the maximum range of service time. The ticket probability exists to keep track of the tickets sold, however they are not going to affect the simulation directly since the simulation focuses on the relationship between time and the counters. If the counter in the simulation is more than one, the user can have a granularity to control each counter whether the counter is initially open or close. The closed counter will automatically change its state to open when there are more than two customers waiting in the queue in any counter opened.

This simulation also has a few formulas for randomization. The program is randomize to allow the simulation to be a little realistic however, for the sake of calculated randomization, The Linear Congruential Generator (LCG) formulas are used.Which are the mixed LCG, additive LCG and the multiplication LCG. The formula is RN = [(Seed*A)+C] mod M , in which A and C is inputted by the user.When the additive is selected, A=1 and when the multiplication LCG is selected, C=0. There will also be a totally random number generator which is a FreeMat randi(x,y) function for the user as an option for a more realistic and unpredictable situation.

The simulation starts with the number of ticket type, number of counters, number of service time type, number of inter-arrival time type, number of customers, and type of randomizer used for the simulation from the user's input. After every input is accepted, the simulation will begin and the step by step simulation is shown . The speed of the step by step is

determined by the speed that the user wants at the beginning of the simulation whether its instant or default. Default will generate the result as if the simulation is taking place in virtual real time. The instant in the other hand will generate everything instantly and show the finished simulation. Other than that, the output will also display the customer summary,counter summary and the simulation summary. The customer summary will be the random interval time generated,the random ticker generated, ticket type, ticket quantity and the amount paid for every iteration.
The counter summary on the other hand, will display the random service time generated, the start and end of the service, the waiting time and the time in the system for every iteration.
The simulation summary will calculate and show the average time,average inter arrival time, average arrival time and the average time spent in the system for every iteration as well as the individual counters average service and their total sales.

Diagrams

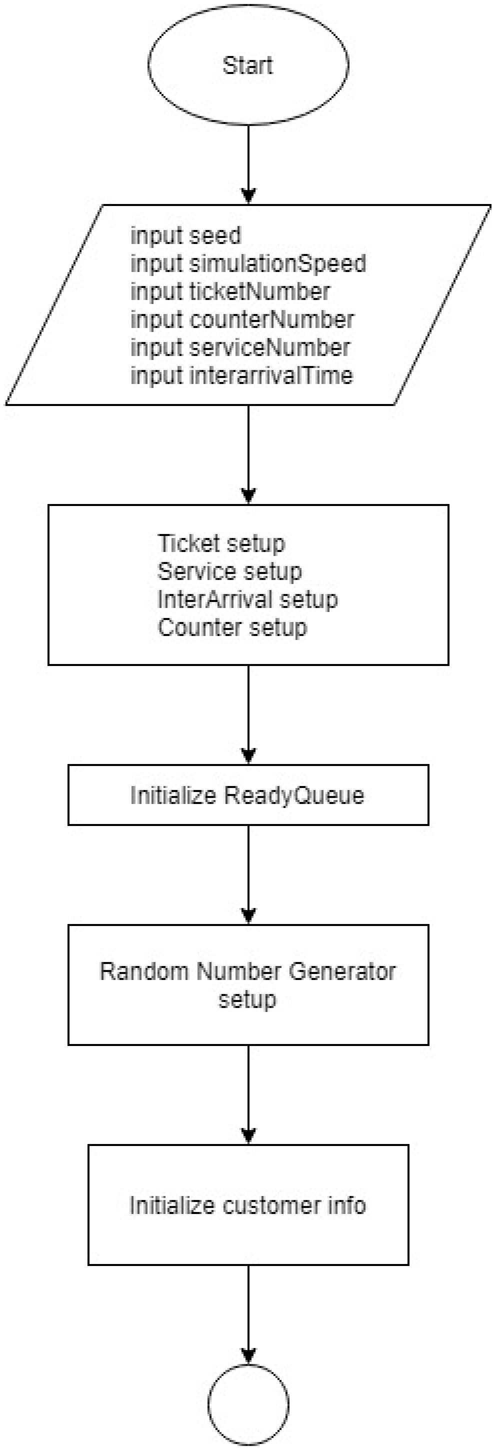


Diagram 2.1

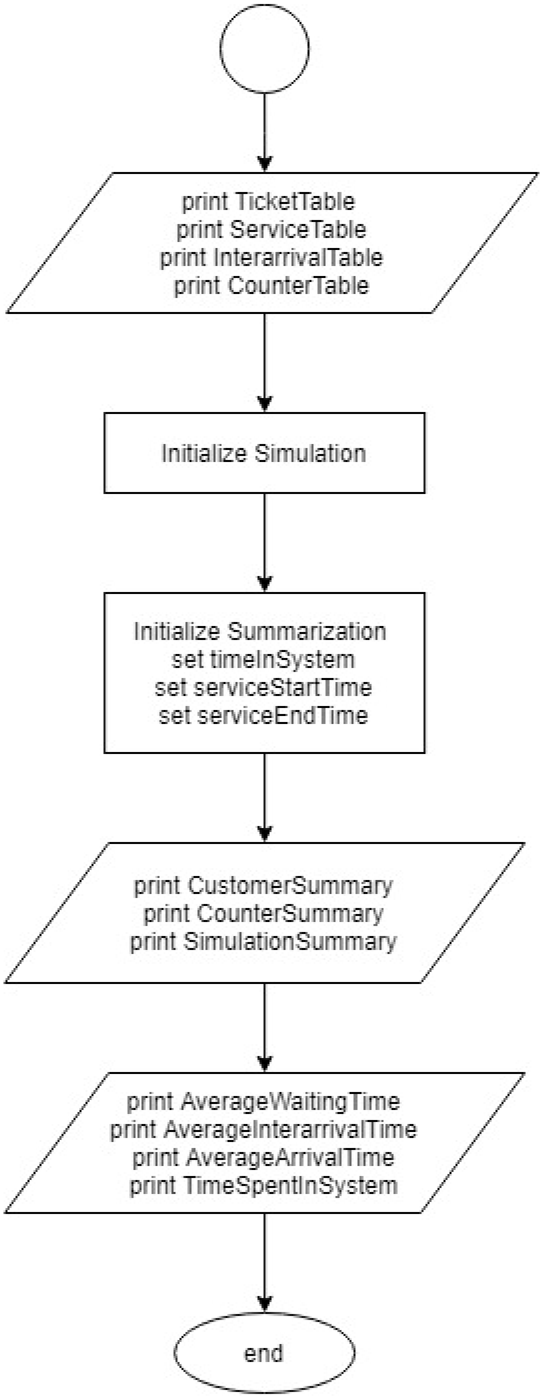
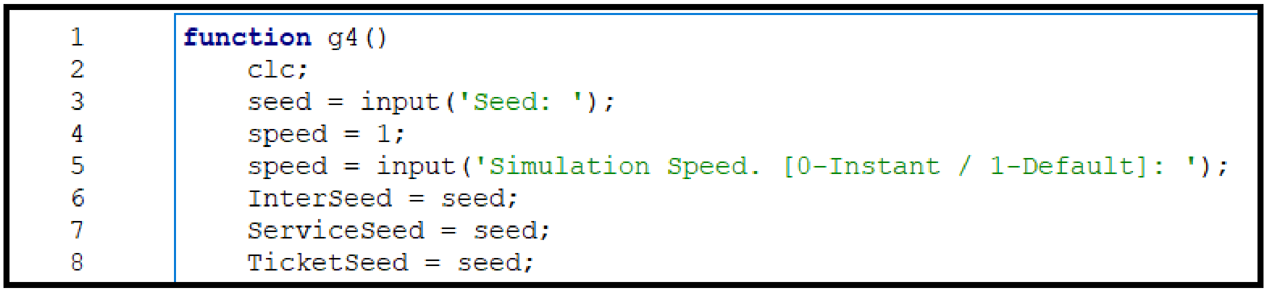

print TicketTable
print ServiceTable
print InterarrivalTable
print CounterTable
Initialize Simulation
Initialize Summarization
set timeInSystem
set serviceStartTime
set serviceEndTime
print CustomerSummary
print CounterSummary
print SimulationSummary
print AverageWaitingTime
print AverageInterarrivalTime
print AverageArrivalTime
print TimeSpentInSystem
end

Diagram 2.2

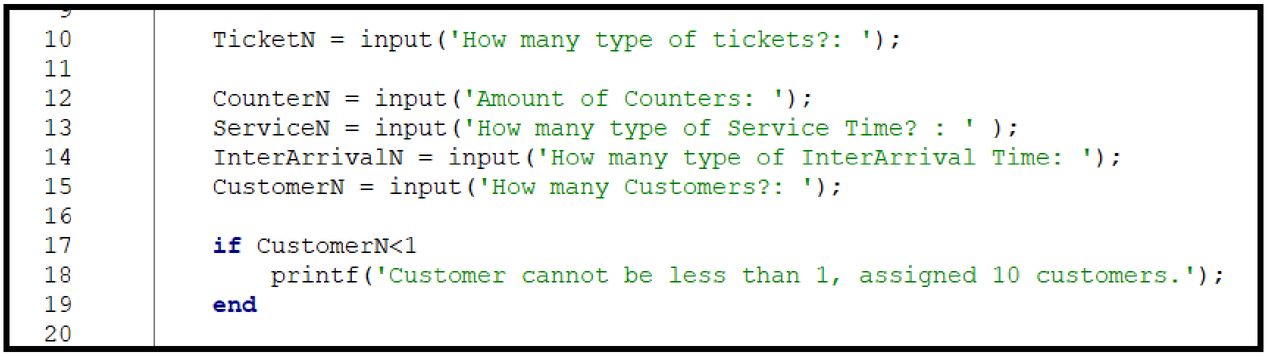
Codes and Algorithms

g4.m (main file)



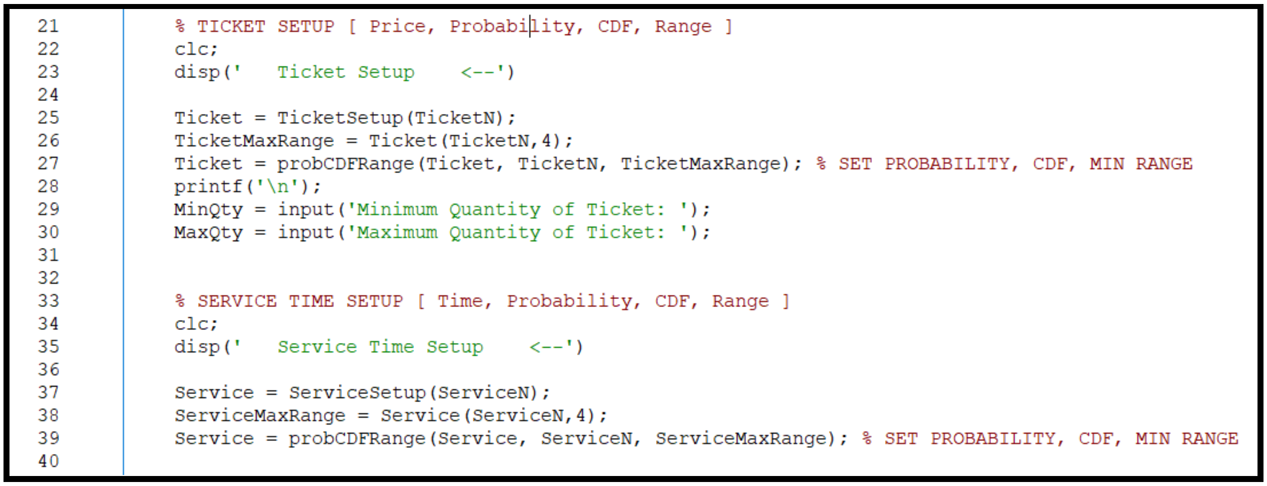
```
function g4()
    clc;
    seed = input('Seed: ');
    speed = 1;
    speed = input('Simulation Speed. [0-Instant / 1-Default]: ');
    InterSeed = seed;
    ServiceSeed = seed;
    TicketSeed = seed;
```

Diagram 3.1 Asks the user to input seeds for generating random numbers for interarrival time, service time and ticket.



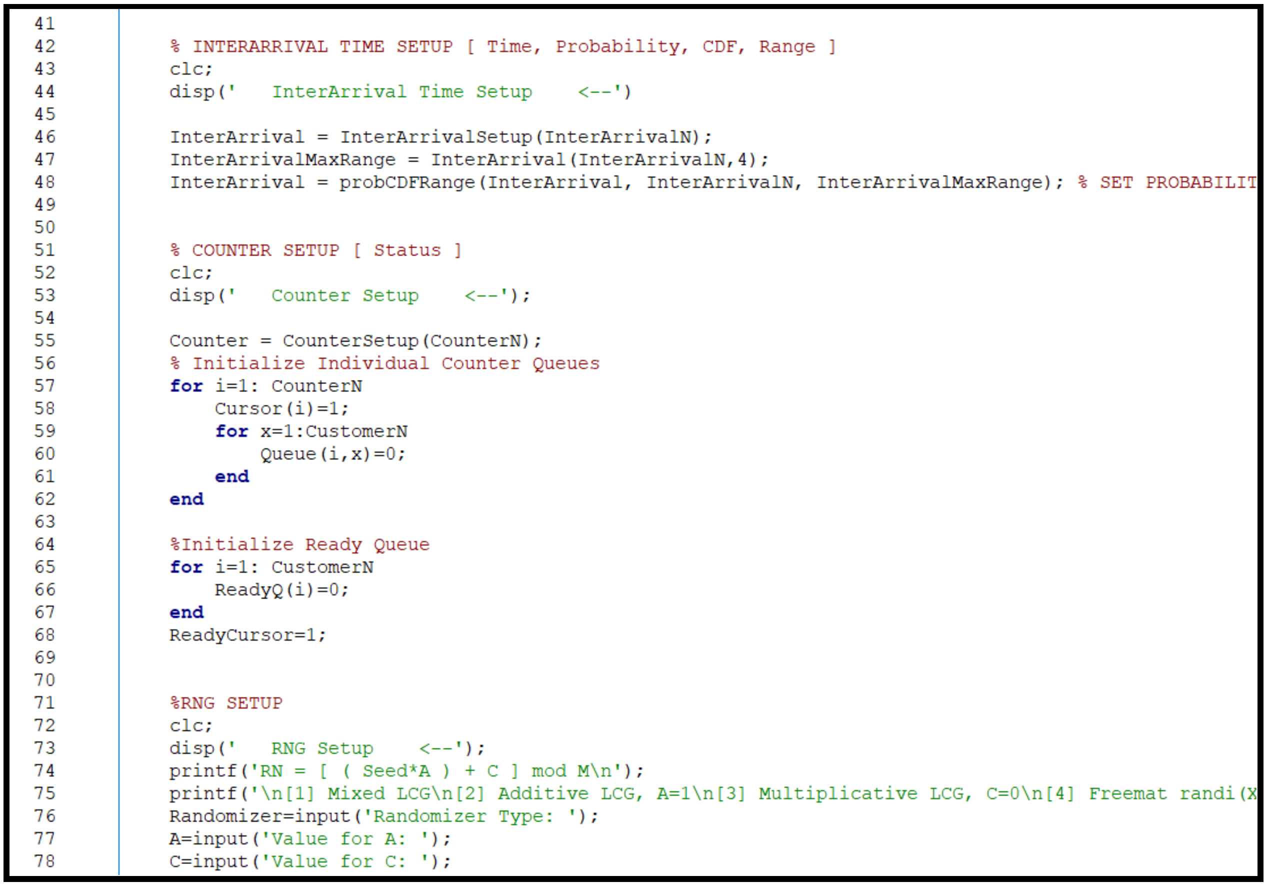
```
    TicketN = input('How many type of tickets?: ');

    CounterN = input('Amount of Counters: ');
    ServiceN = input('How many type of Service Time? : ' );
    InterArrivalN = input('How many type of InterArrival Time: ');
    CustomerN = input('How many Customers?: ');

    if CustomerN<1
        printf('Customer cannot be less than 1, assigned 10 customers.');
    end

```

Diagram 3.2 Here it asks the user to input the number of tickets, counters, service type, interarrival time and customer.



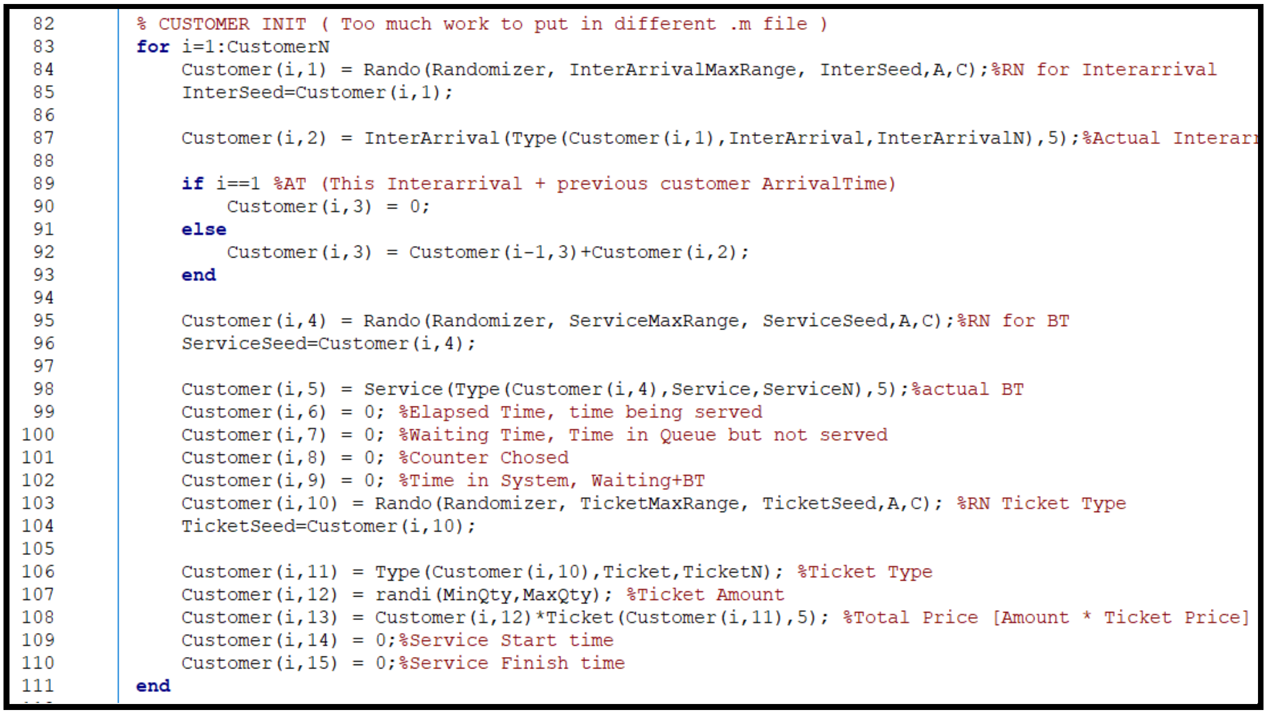
```
    % TICKET SETUP [ Price, Probability, CDF, Range ]
    clc;
    disp('   Ticket Setup    <--')

    Ticket = TicketSetup(TicketN);
    TicketMaxRange = Ticket(TicketN,4);
    Ticket = probCDFRange(Ticket, TicketN, TicketMaxRange); % SET PROBABILITY, CDF, MIN RANGE
    printf('\n');
    MinQty = input('Minimum Quantity of Ticket: ');
    MaxQty = input('Maximum Quantity of Ticket: ');


    % SERVICE TIME SETUP [ Time, Probability, CDF, Range ]
    clc;
    disp('   Service Time Setup    <--')

    Service = ServiceSetup(ServiceN);
    ServiceMaxRange = Service(ServiceN,4);
    Service = probCDFRange(Service, ServiceN, ServiceMaxRange); % SET PROBABILITY, CDF, MIN RANGE

```

Diagram 3.3 Setting up Ticket and Service Time



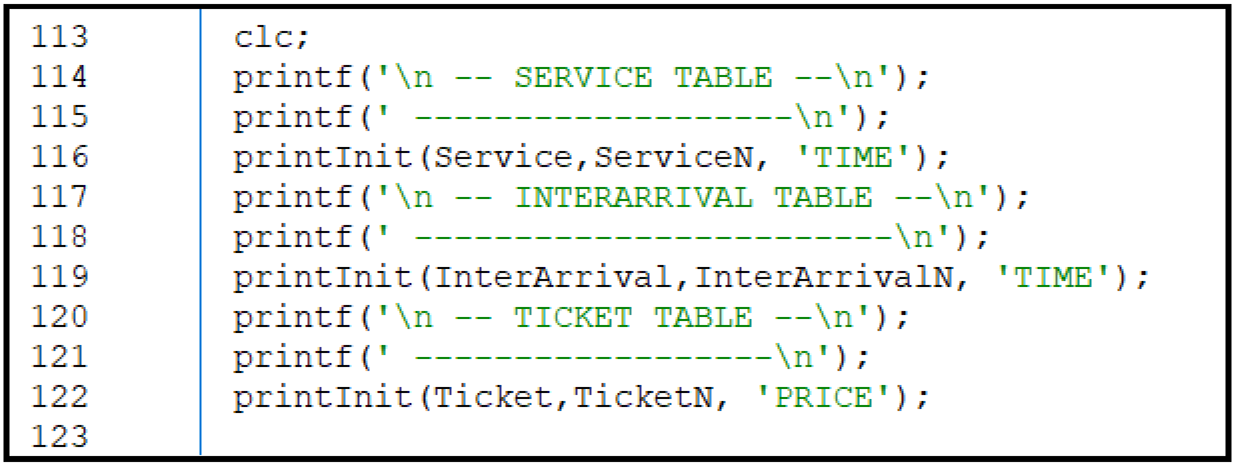
```

    % INTERARRIVAL TIME SETUP [ Time, Probability, CDF, Range ]
    clc;
    disp('   InterArrival Time Setup    <--')

    InterArrival = InterArrivalSetup(InterArrivalN);
    InterArrivalMaxRange = InterArrival(InterArrivalN,4);
    InterArrival = probCDFRange(InterArrival, InterArrivalN, InterArrivalMaxRange); % SET PROBABILIT


    % COUNTER SETUP [ Status ]
    clc;
    disp('   Counter Setup    <--');

    Counter = CounterSetup(CounterN);
    % Initialize Individual Counter Queues
    for i=1: CounterN
        Cursor(i)=1;
        for x=1:CustomerN
            Queue(i,x)=0;
        end
    end

    %Initialize Ready Queue
    for i=1: CustomerN
        ReadyQ(i)=0;
    end
    ReadyCursor=1;


    %RNG SETUP
    clc;
    disp('   RNG Setup    <--');
    printf('RN = [ ( Seed*A ) + C ] mod M\n');
    printf('\n[1] Mixed LCG\n[2] Additive LCG, A=1\n[3] Multiplicative LCG, C=0\n[4] Freemat randi(X
    Randomizer=input('Randomizer Type: ');
    A=input('Value for A: ');
    C=input('Value for C: ');
```

Diagram 3.4 Setting up Interarrival Time, Counter, RNG and initializing Ready Queue



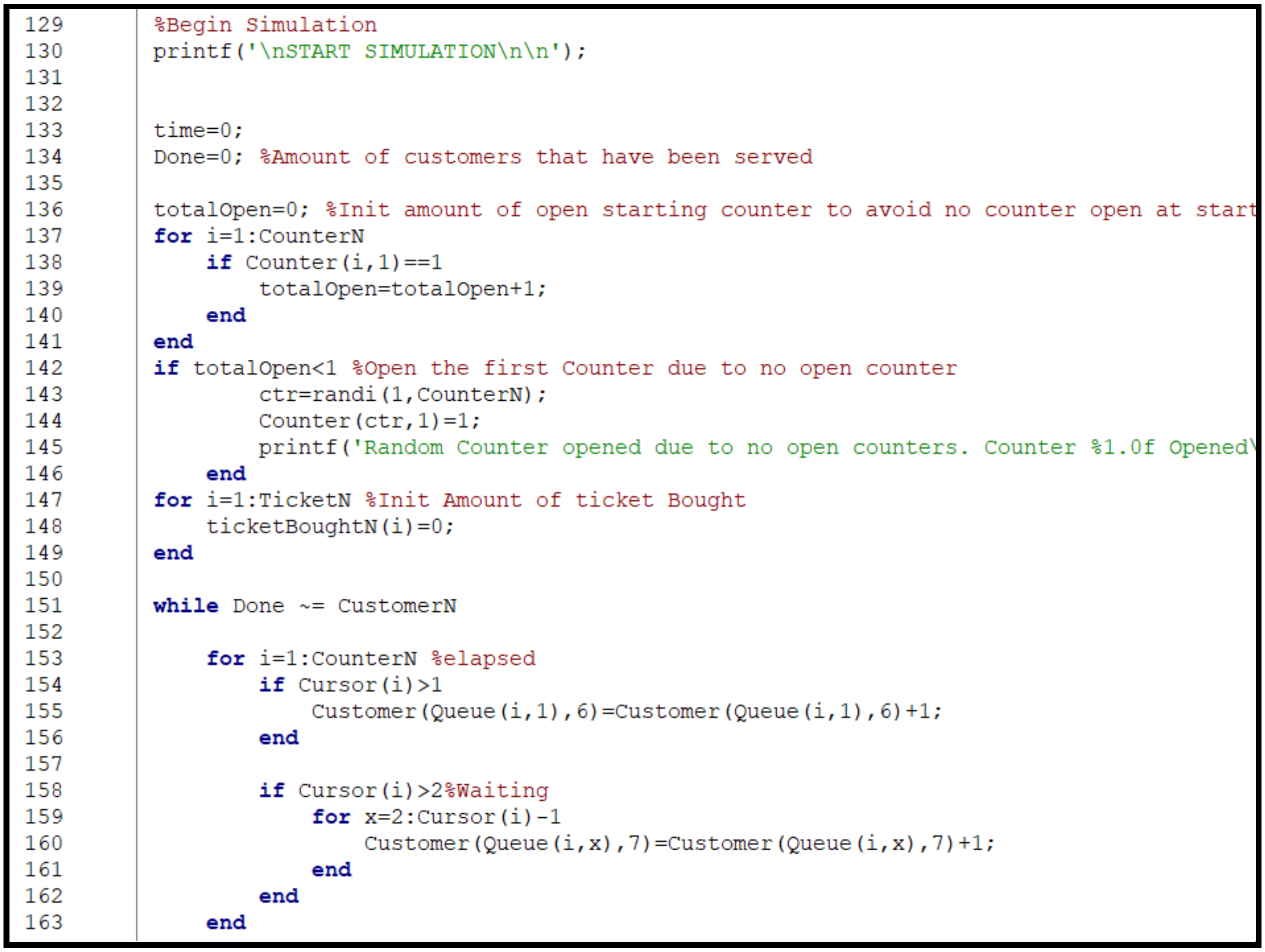
```
% CUSTOMER INIT ( Too much work to put in different .m file )
for i=1:CustomerN
    Customer(i,1) = Rando(Randomizer, InterArrivalMaxRange, InterSeed,A,C);%RN for Interarrival
    InterSeed=Customer(i,1);

    Customer(i,2) = InterArrival(Type(Customer(i,1),InterArrival,InterArrivalN),5);%Actual Interar

    if i==1 %AT (This Interarrival + previous customer ArrivalTime)
        Customer(i,3) = 0;
    else
        Customer(i,3) = Customer(i-1,3)+Customer(i,2);
    end

    Customer(i,4) = Rando(Randomizer, ServiceMaxRange, ServiceSeed,A,C);%RN for BT
    ServiceSeed=Customer(i,4);

    Customer(i,5) = Service(Type(Customer(i,4),Service,ServiceN),5);%actual BT
    Customer(i,6) = 0; %Elapsed Time, time being served
    Customer(i,7) = 0; %Waiting Time, Time in Queue but not served
    Customer(i,8) = 0; %Counter Chosed
    Customer(i,9) = 0; %Time in System, Waiting+BT
    Customer(i,10) = Rando(Randomizer, TicketMaxRange, TicketSeed,A,C); %RN Ticket Type
    TicketSeed=Customer(i,10);

    Customer(i,11) = Type(Customer(i,10),Ticket,TicketN); %Ticket Type
    Customer(i,12) = randi(MinQty,MaxQty); %Ticket Amount
    Customer(i,13) = Customer(i,12)*Ticket(Customer(i,11),5); %Total Price [Amount * Ticket Price]
    Customer(i,14) = 0;%Service Start time
    Customer(i,15) = 0;%Service Finish time
end
```

Diagram 3.5 Initialize Customer function



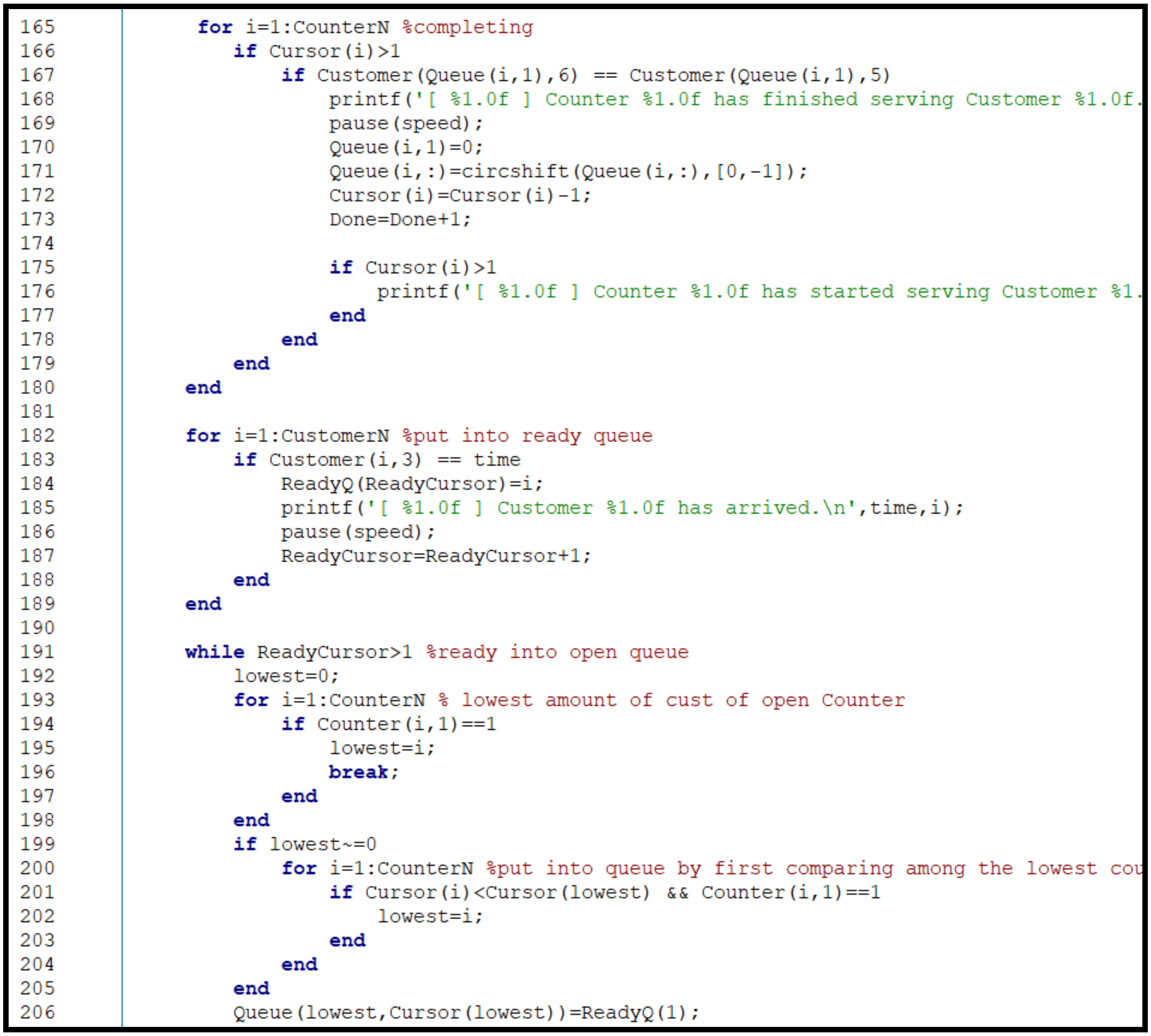
```
clc;
printf('\n -- SERVICE TABLE --\n');
printf(' --------------------\n');
printInit(Service,ServiceN, 'TIME');
printf('\n -- INTERARRIVAL TABLE --\n');
printf(' -------------------------\n');
printInit(InterArrival,InterArrivalN, 'TIME');
printf('\n -- TICKET TABLE --\n');
printf(' -------------------\n');
printInit(Ticket,TicketN, 'PRICE');
```

Diagram 3.6 Displays Service Table, Interarrival Table and Ticket Table



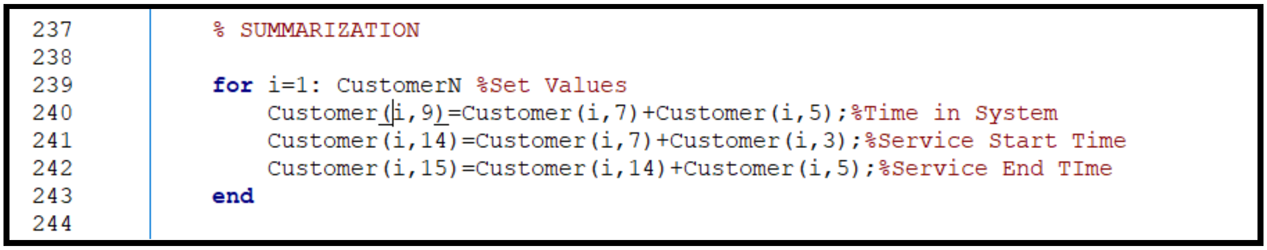
```
%Begin Simulation
printf('\nSTART SIMULATION\n\n');


time=0;
Done=0; %Amount of customers that have been served

totalOpen=0; %Init amount of open starting counter to avoid no counter open at start
for i=1:CounterN
    if Counter(i,1)==1
        totalOpen=totalOpen+1;
    end
end
if totalOpen<1 %Open the first Counter due to no open counter
        ctr=randi(1,CounterN);
        Counter(ctr,1)=1;
        printf('Random Counter opened due to no open counters. Counter %1.0f Opened\
    end
for i=1:TicketN %Init Amount of ticket Bought
    ticketBoughtN(i)=0;
end

while Done ~= CustomerN

    for i=1:CounterN %elapsed
        if Cursor(i)>1
            Customer(Queue(i,1),6)=Customer(Queue(i,1),6)+1;
        end

        if Cursor(i)>2%Waiting
            for x=2:Cursor(i)-1
                Customer(Queue(i,x),7)=Customer(Queue(i,x),7)+1;
            end
        end
    end
```

Diagram 3.7 Simulation execution



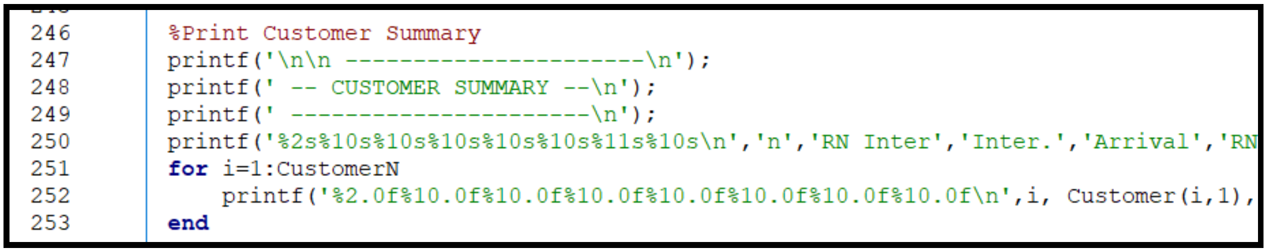
```
 for i=1:CounterN %completing
    if Cursor(i)>1
        if Customer(Queue(i,1),6) == Customer(Queue(i,1),5)
            printf('[ %1.0f ] Counter %1.0f has finished serving Customer %1.0f.
            pause(speed);
            Queue(i,1)=0;
            Queue(i,:)=circshift(Queue(i,:),[0,-1]);
            Cursor(i)=Cursor(i)-1;
            Done=Done+1;

            if Cursor(i)>1
                printf('[ %1.0f ] Counter %1.0f has started serving Customer %1.
            end
        end
    end
end

for i=1:CustomerN %put into ready queue
    if Customer(i,3) == time
        ReadyQ(ReadyCursor)=i;
        printf('[ %1.0f ] Customer %1.0f has arrived.\n',time,i);
        pause(speed);
        ReadyCursor=ReadyCursor+1;
    end
end

while ReadyCursor>1 %ready into open queue
    lowest=0;
    for i=1:CounterN % lowest amount of cust of open Counter
        if Counter(i,1)==1
            lowest=i;
            break;
        end
    end
    if lowest~=0
        for i=1:CounterN %put into queue by first comparing among the lowest cou
            if Cursor(i)<Cursor(lowest) && Counter(i,1)==1
                lowest=i;
            end
        end
    end
    Queue(lowest,Cursor(lowest))=ReadyQ(1);
```

Diagram 3.8 Simulation execution



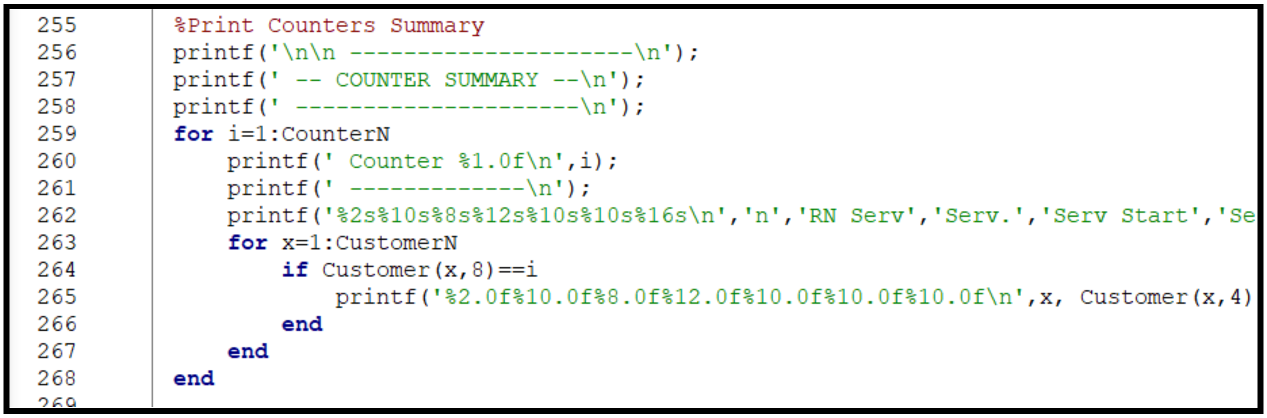
```
% SUMMARIZATION

for i=1: CustomerN %Set Values
    Customer(i,9)=Customer(i,7)+Customer(i,5);%Time in System
    Customer(i,14)=Customer(i,7)+Customer(i,3);%Service Start Time
    Customer(i,15)=Customer(i,14)+Customer(i,5);%Service End TIme
end

```

Diagram 3.9 Setting values for summarization



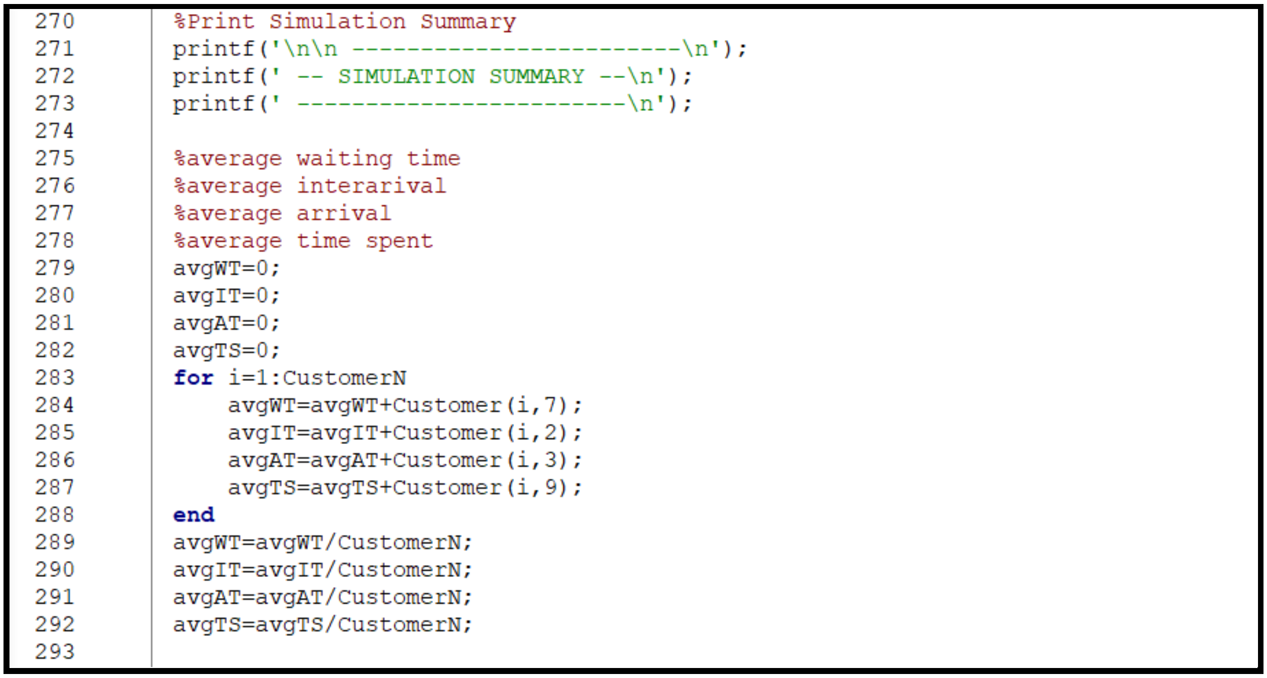
```
%Print Customer Summary
printf('\n\n ----------------------\n');
printf(' -- CUSTOMER SUMMARY --\n');
printf(' ----------------------\n');
printf('%2s%10s%10s%10s%10s%10s%11s%10s\n','n','RN Inter','Inter.','Arrival','RN
for i=1:CustomerN
    printf('%2.0f%10.0f%10.0f%10.0f%10.0f%10.0f%10.0f%10.0f\n',i, Customer(i,1),
end
```

Diagram 3.10 Display Customer Summary, refer code for full lines



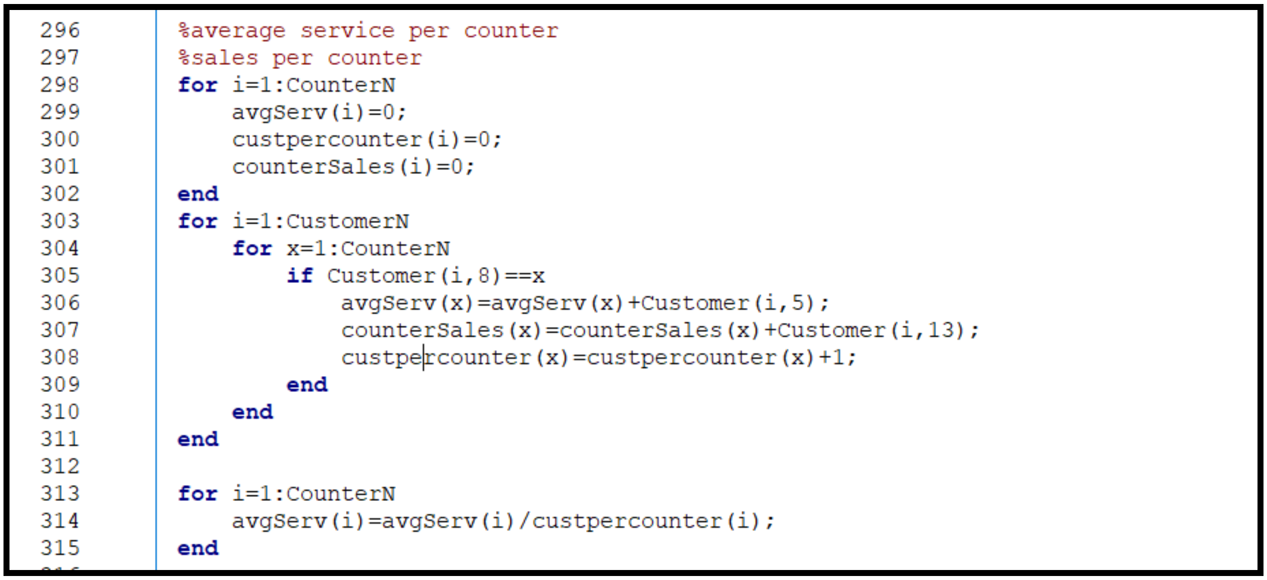
```
%Print Counters Summary
printf('\n\n ---------------------\n');
printf(' -- COUNTER SUMMARY --\n');
printf(' ---------------------\n');
for i=1:CounterN
    printf(' Counter %1.0f\n',i);
    printf(' -------------\n');
    printf('%2s%10s%8s%12s%10s%10s%16s\n','n','RN Serv','Serv.','Serv Start','Se
    for x=1:CustomerN
        if Customer(x,8)==i
            printf('%2.0f%10.0f%8.0f%12.0f%10.0f%10.0f%10.0f\n',x, Customer(x,4)
        end
    end
end
```

Diagram 3.11 Display Counter Summary, refer code for full line



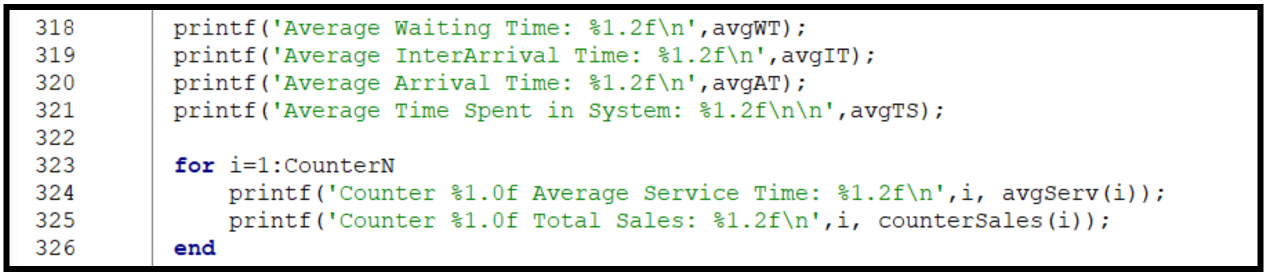
```
%Print Simulation Summary
printf('\n\n ------------------------\n');
printf(' -- SIMULATION SUMMARY --\n');
printf(' ------------------------\n');

%average waiting time
%average interarival
%average arrival
%average time spent
avgWT=0;
avgIT=0;
avgAT=0;
avgTS=0;
for i=1:CustomerN
    avgWT=avgWT+Customer(i,7);
    avgIT=avgIT+Customer(i,2);
    avgAT=avgAT+Customer(i,3);
    avgTS=avgTS+Customer(i,9);
end
avgWT=avgWT/CustomerN;
avgIT=avgIT/CustomerN;
avgAT=avgAT/CustomerN;
avgTS=avgTS/CustomerN;
```

Diagram 3.12 Getting average waiting, interarrival, arrival and spent time for simulation summary



```
%average service per counter
%sales per counter
for i=1:CounterN
    avgServ(i)=0;
    custpercounter(i)=0;
    counterSales(i)=0;
end
for i=1:CustomerN
    for x=1:CounterN
        if Customer(i,8)==x
            avgServ(x)=avgServ(x)+Customer(i,5);
            counterSales(x)=counterSales(x)+Customer(i,13);
            custpercounter(x)=custpercounter(x)+1;
        end
    end
end

for i=1:CounterN
    avgServ(i)=avgServ(i)/custpercounter(i);
end
```

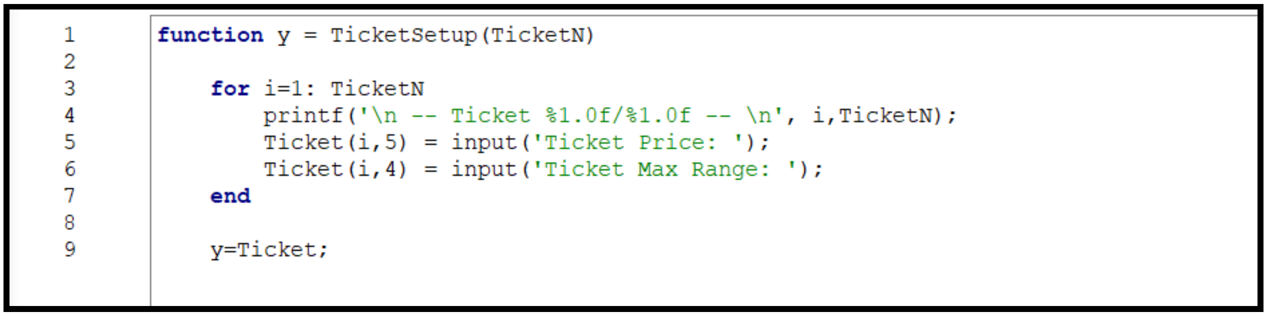
Diagram 3.13 Get average service time per counter



```
printf('Average Waiting Time: %1.2f\n',avgWT);
printf('Average InterArrival Time: %1.2f\n',avgIT);
printf('Average Arrival Time: %1.2f\n',avgAT);
printf('Average Time Spent in System: %1.2f\n\n',avgTS);

for i=1:CounterN
    printf('Counter %1.0f Average Service Time: %1.2f\n',i, avgServ(i));
    printf('Counter %1.0f Total Sales: %1.2f\n',i, counterSales(i));
end
```

Diagram 3.14 Displays all average

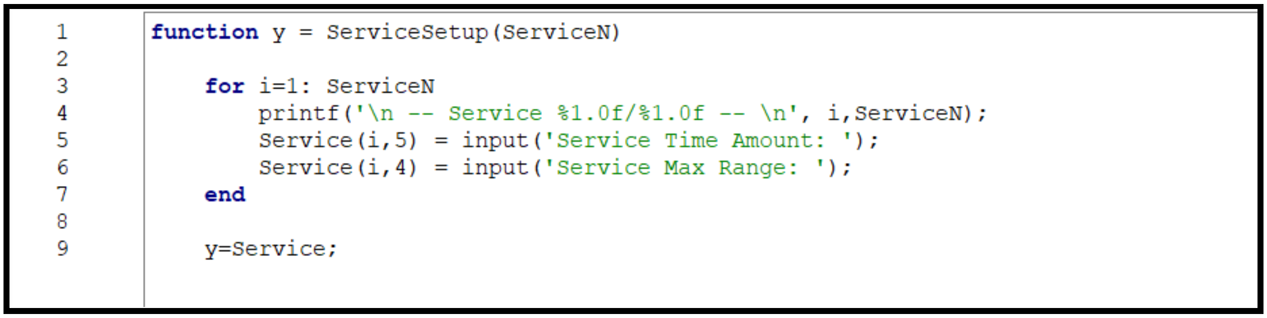
TicketSetup.m



```
function y = TicketSetup(TicketN)

    for i=1: TicketN
        printf('\n -- Ticket %1.0f/%1.0f -- \n', i,TicketN);
        Ticket(i,5) = input('Ticket Price: ');
        Ticket(i,4) = input('Ticket Max Range: ');
    end

    y=Ticket;
```

Diagram 3.15 Ticket Setup function

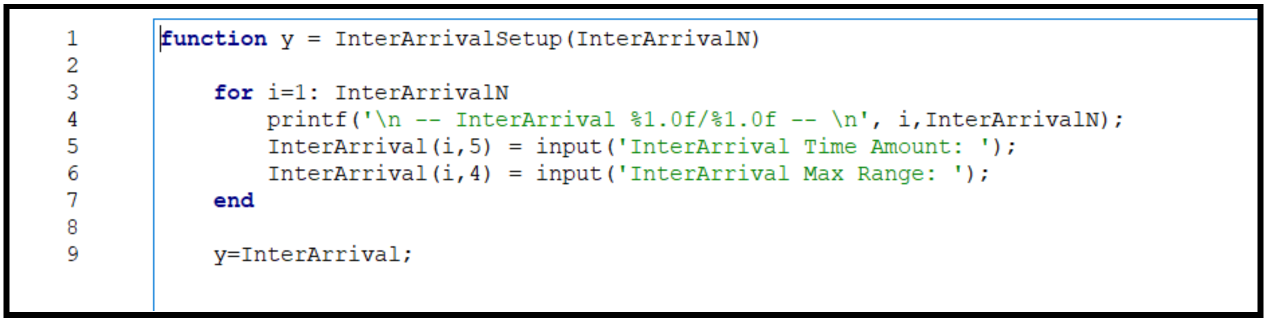
ServiceSetup.m



```
function y = ServiceSetup(ServiceN)

    for i=1: ServiceN
        printf('\n -- Service %1.0f/%1.0f -- \n', i,ServiceN);
        Service(i,5) = input('Service Time Amount: ');
        Service(i,4) = input('Service Max Range: ');
    end

    y=Service;
```

Diagram 3.16 Service Setup function

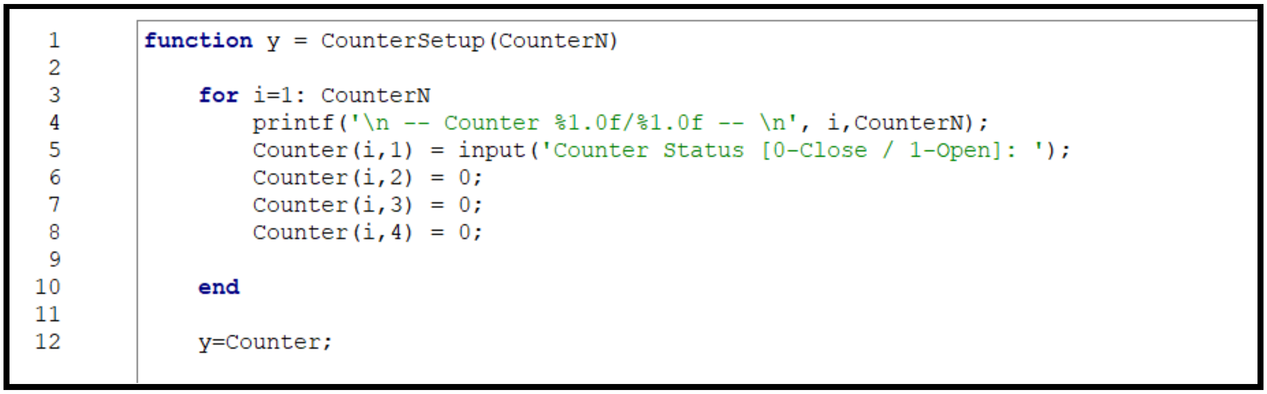
InterArrivalSetup.m



```
function y = InterArrivalSetup(InterArrivalN)

    for i=1: InterArrivalN
        printf('\n -- InterArrival %1.0f/%1.0f -- \n', i,InterArrivalN);
        InterArrival(i,5) = input('InterArrival Time Amount: ');
        InterArrival(i,4) = input('InterArrival Max Range: ');
    end

    y=InterArrival;
```

Diagram 3.17 Interarrival Setup function

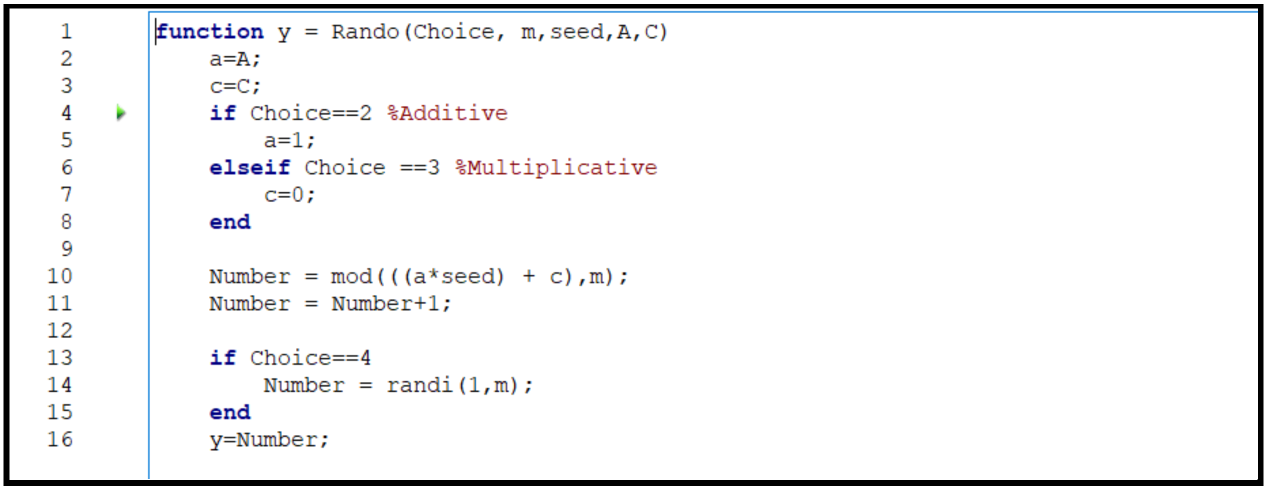
CounterSetup.m



```
function y = CounterSetup(CounterN)

    for i=1: CounterN
        printf('\n -- Counter %1.0f/%1.0f -- \n', i,CounterN);
        Counter(i,1) = input('Counter Status [0-Close / 1-Open]: ');
        Counter(i,2) = 0;
        Counter(i,3) = 0;
        Counter(i,4) = 0;

    end

    y=Counter;
```

Diagram 3.18 Counter Setup function

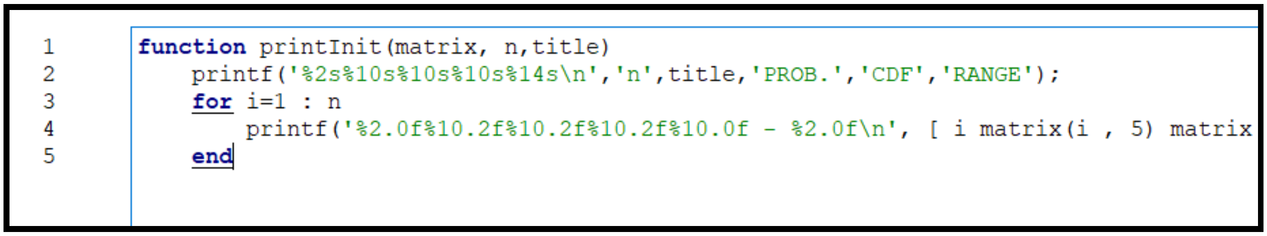
Rando.m



```
function y = Rando(Choice, m,seed,A,C)
    a=A;
    c=C;
    if Choice==2 %Additive
        a=1;
    elseif Choice ==3 %Multiplicative
        c=0;
    end

    Number = mod(((a*seed) + c),m);
    Number = Number+1;

    if Choice==4
        Number = randi(1,m);
    end
    y=Number;
```

Diagram 3.19 Randomizer function

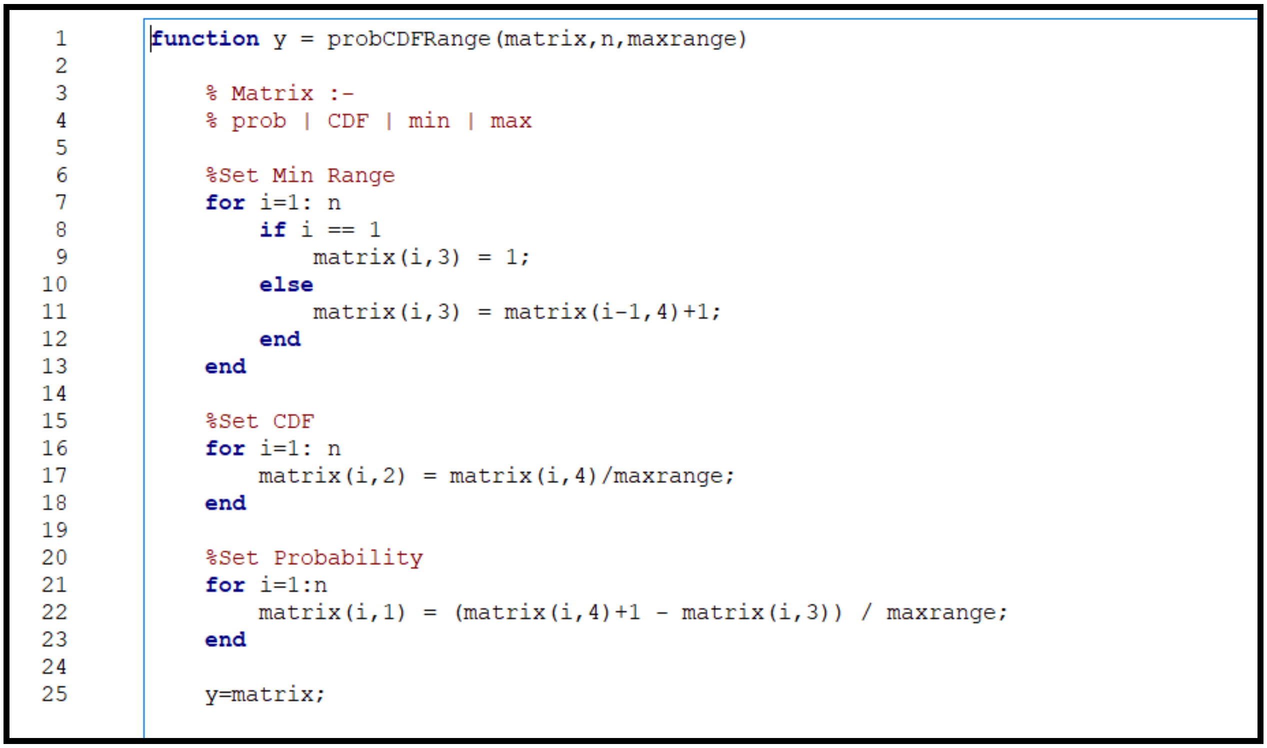
printInit.m



```
function printInit(matrix, n,title)
    printf('%2s%10s%10s%10s%14s\n','n',title,'PROB.','CDF','RANGE');
    for i=1 : n
        printf('%2.0f%10.2f%10.2f%10.2f%10.0f - %2.0f\n', [ i matrix(i , 5) matrix
    end
```

Diagram 3.20 Print function for displaying data in table form (refer code for full line)

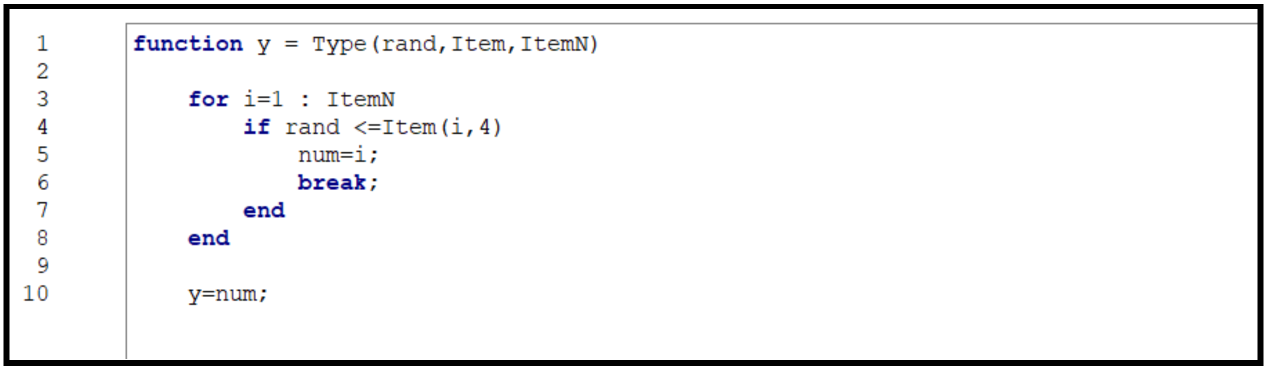
probCDFRange.m



```
function y = probCDFRange(matrix,n,maxrange)

    % Matrix :-
    % prob | CDF | min | max

    %Set Min Range
    for i=1: n
        if i == 1
            matrix(i,3) = 1;
        else
            matrix(i,3) = matrix(i-1,4)+1;
        end
    end

    %Set CDF
    for i=1: n
        matrix(i,2) = matrix(i,4)/maxrange;
    end

    %Set Probability
    for i=1:n
        matrix(i,1) = (matrix(i,4)+1 - matrix(i,3)) / maxrange;
    end

    y=matrix;
```

Diagram 3.21 Function for setting probability, CDF and min range

Type.m



```
function y = Type(rand,Item,ItemN)

    for i=1 : ItemN
        if rand <=Item(i,4)
            num=i;
            break;
        end
    end

    y=num;
```

Diagram 3.22 Function to get value by comparing random generated number with item number range

Algorithm

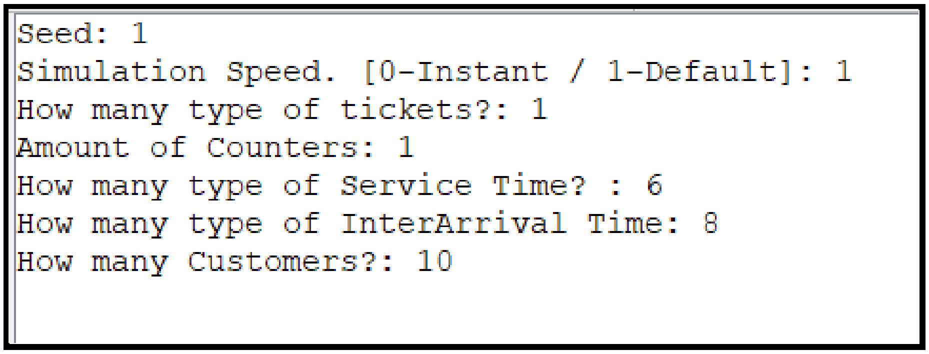
```
Start
input seed
input simulation speed
.input ticketNumber
input counterNumber
input serviceNumber
input interarrivalTime
set Ticket;
set Service;
set InterArrival;
set Counter;
Initialize ReadyQueue;
set RNG;
Initialize Customer;
print TicketTable;
print ServiceTable;
print InterarrivalTable;
print CounterTable;
Initialize Simulation;
Initialize Summarization;
set timeInSystem;
set serviceStartTime;
set serviceEndTime;
print CustomerSummary;
print CounterSummary;
print SimulationSummary
print AverageWaitingTime;
print AverageInterarrivalTime;
print AverageArrivalTime;
print TimeSpentInSystem;
end
```

Input and Output of Simulation

Input and Output of 1 Counter

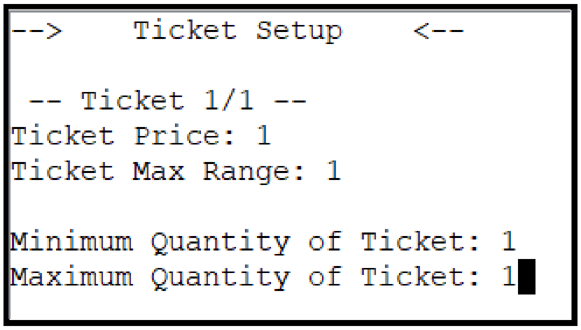
For this simulation we have decided to use the value of 1 as the seed number. Besides that, for random number generators we used mixed LCG with the value for ‘a’ is 4 and the value of ‘c’ is 3.

Input :



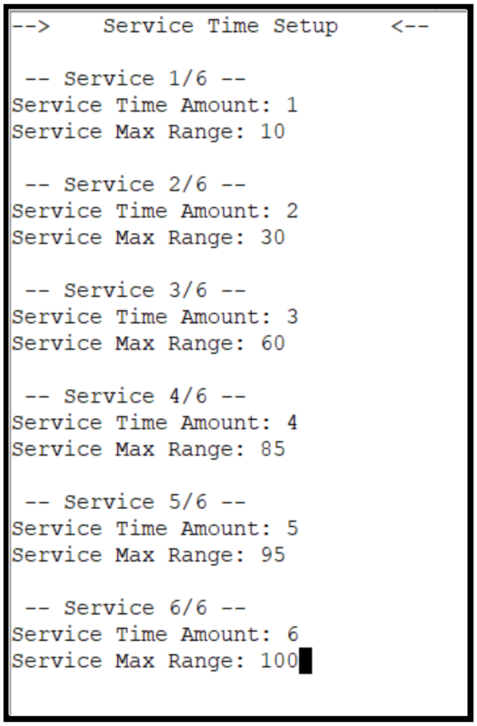
```
Seed: 1
Simulation Speed. [0-Instant / 1-Default]: 1
How many type of tickets?: 1
Amount of Counters: 1
How many type of Service Time? : 6
How many type of InterArrival Time: 8
How many Customers?: 10
```

Diagram 4.1 Initialize the variable needed for simulation



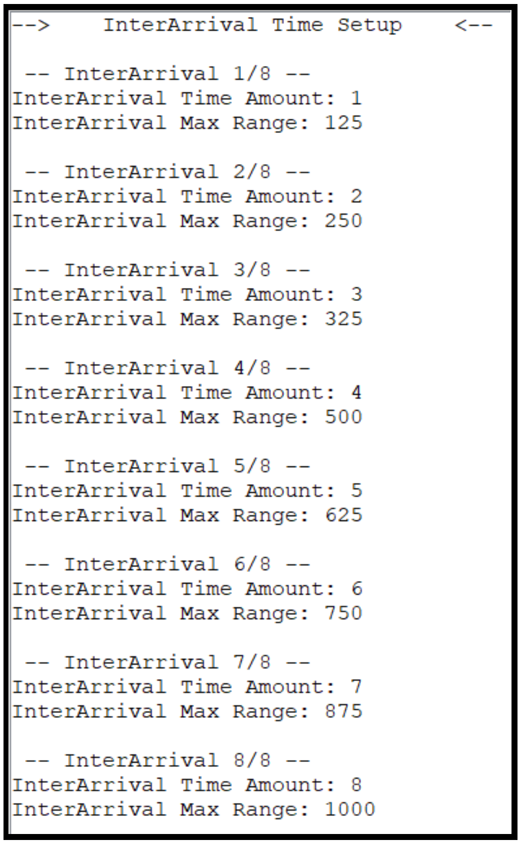
```
-->    Ticket Setup    <--

 -- Ticket 1/1 --
Ticket Price: 1
Ticket Max Range: 1

Minimum Quantity of Ticket: 1
Maximum Quantity of Ticket: 1
```

Diagram 4.2 Initialize the price and quantity of ticket



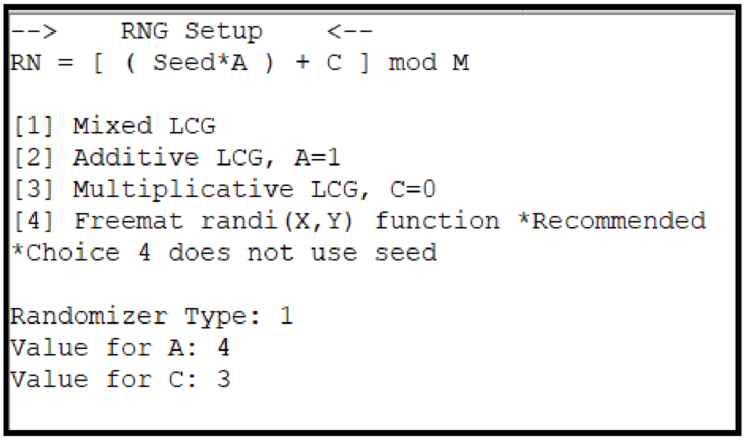
```
-->    Service Time Setup    <--

 -- Service 1/6 --
Service Time Amount: 1
Service Max Range: 10

 -- Service 2/6 --
Service Time Amount: 2
Service Max Range: 30

 -- Service 3/6 --
Service Time Amount: 3
Service Max Range: 60

 -- Service 4/6 --
Service Time Amount: 4
Service Max Range: 85

 -- Service 5/6 --
Service Time Amount: 5
Service Max Range: 95

 -- Service 6/6 --
Service Time Amount: 6
Service Max Range: 100
```

Diagram 4.3 Initialize the value for service time table



```
-->    InterArrival Time Setup    <--

 -- InterArrival 1/8 --
InterArrival Time Amount: 1
InterArrival Max Range: 125

 -- InterArrival 2/8 --
InterArrival Time Amount: 2
InterArrival Max Range: 250

 -- InterArrival 3/8 --
InterArrival Time Amount: 3
InterArrival Max Range: 325

 -- InterArrival 4/8 --
InterArrival Time Amount: 4
InterArrival Max Range: 500

 -- InterArrival 5/8 --
InterArrival Time Amount: 5
InterArrival Max Range: 625

 -- InterArrival 6/8 --
InterArrival Time Amount: 6
InterArrival Max Range: 750

 -- InterArrival 7/8 --
InterArrival Time Amount: 7
InterArrival Max Range: 875

 -- InterArrival 8/8 --
InterArrival Time Amount: 8
InterArrival Max Range: 1000
```

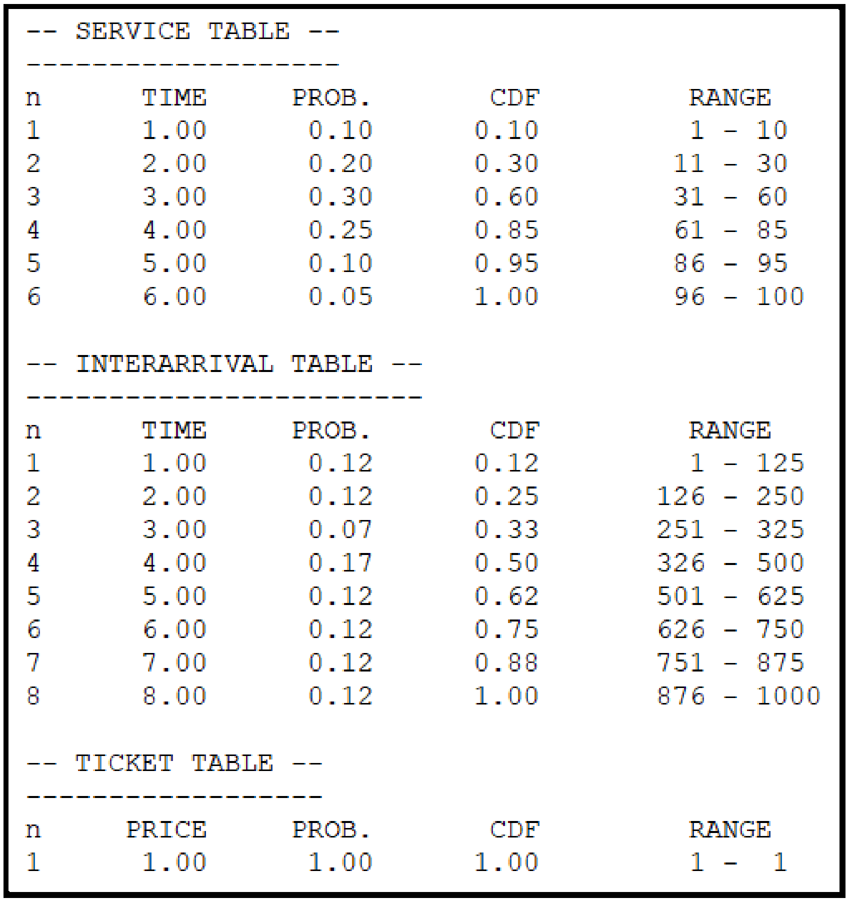
Diagram 4.4 Initialize the value for interarrival time table



```
-->    RNG Setup    <--
RN = [ ( Seed*A ) + C ] mod M

[1] Mixed LCG
[2] Additive LCG, A=1
[3] Multiplicative LCG, C=0
[4] Freemat randi(X,Y) function *Recommended
*Choice 4 does not use seed

Randomizer Type: 1
Value for A: 4
Value for C: 3
```

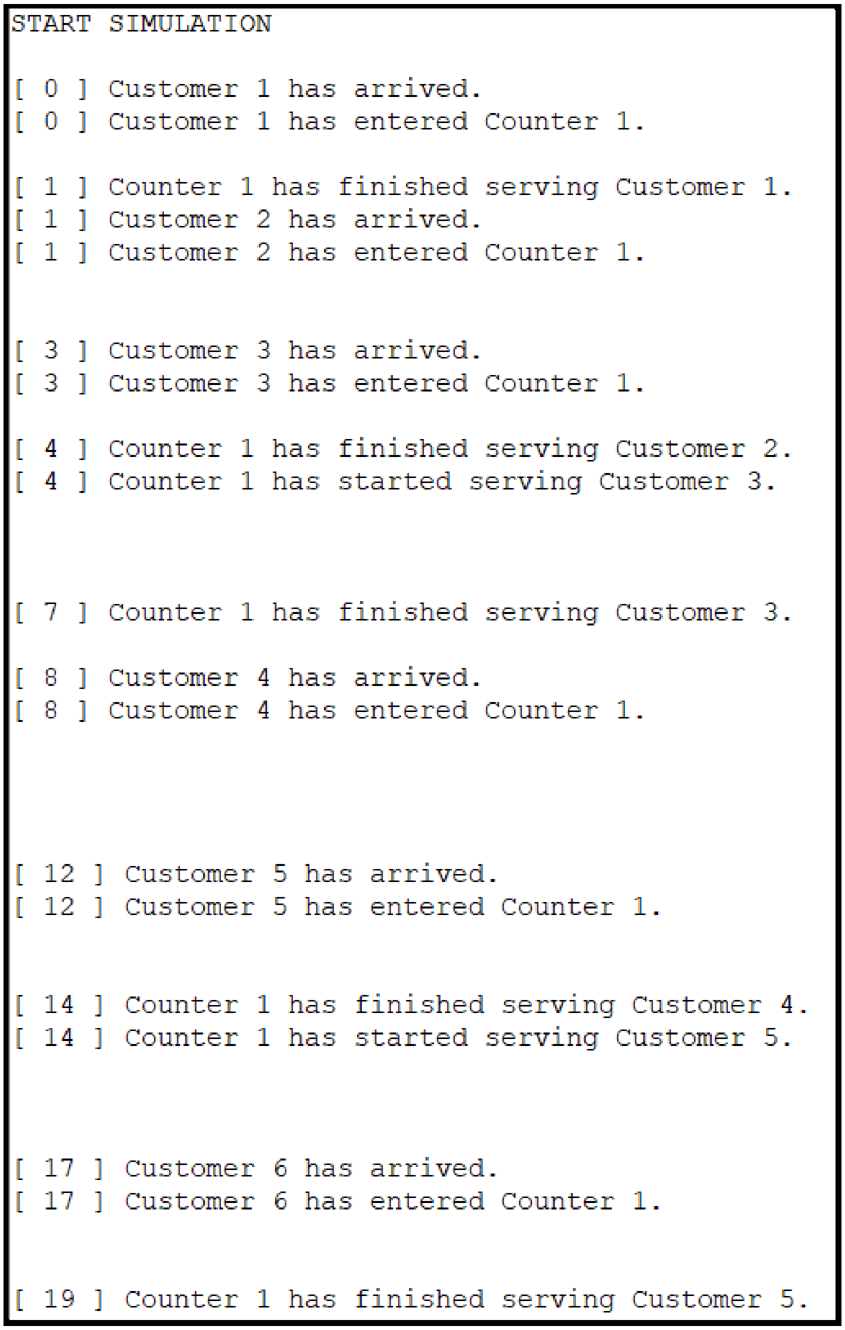
Diagram 4.5 Choose the Random Number Generator Setup

Output :



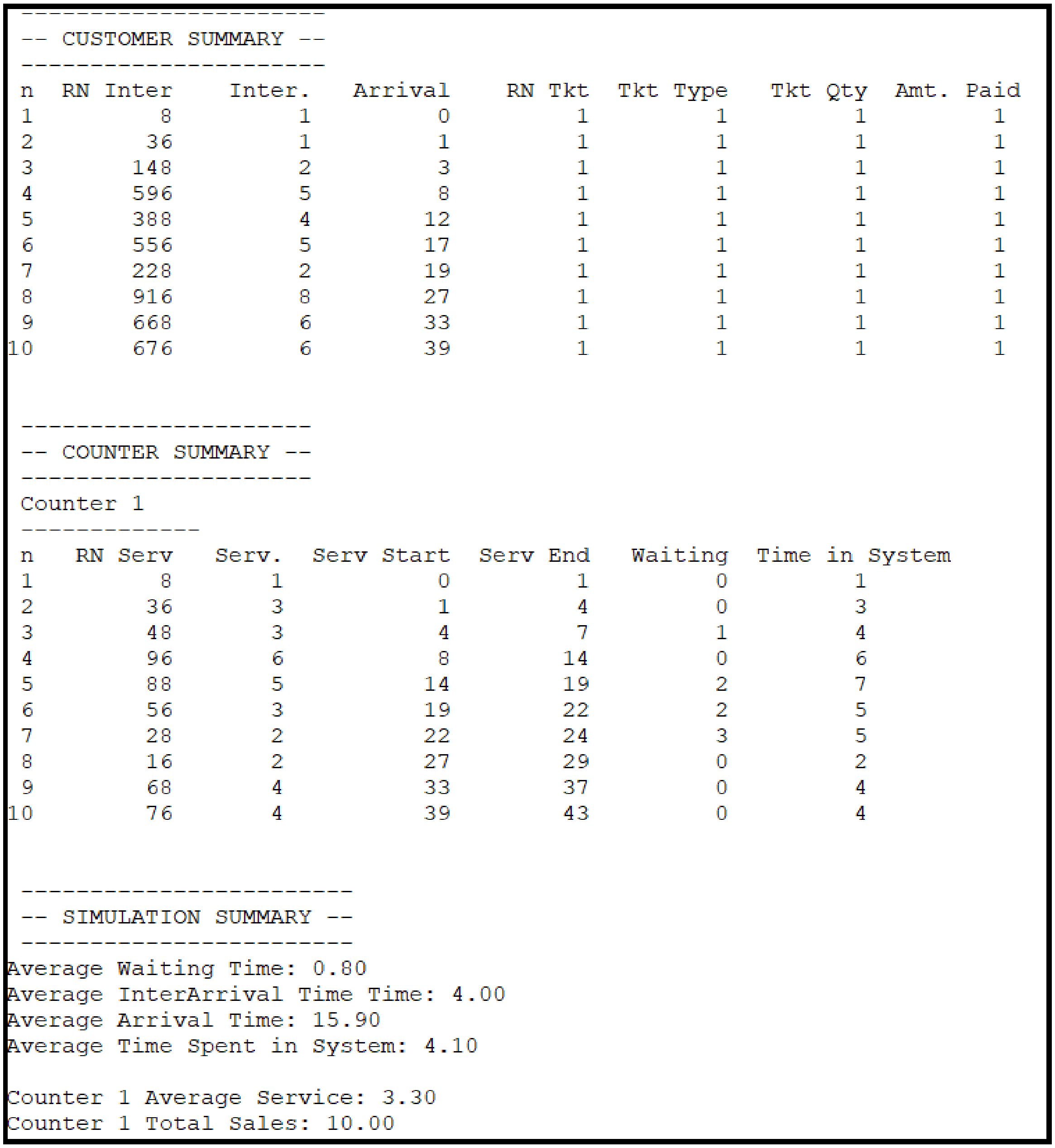
```
-- SERVICE TABLE --
-------------------
n       TIME     PROB.      CDF         RANGE
1       1.00      0.10      0.10         1 - 10
2       2.00      0.20      0.30        11 - 30
3       3.00      0.30      0.60        31 - 60
4       4.00      0.25      0.85        61 - 85
5       5.00      0.10      0.95        86 - 95
6       6.00      0.05      1.00        96 - 100

-- INTERARRIVAL TABLE --
------------------------
n       TIME     PROB.      CDF         RANGE
1       1.00      0.12      0.12         1 - 125
2       2.00      0.12      0.25       126 - 250
3       3.00      0.07      0.33       251 - 325
4       4.00      0.17      0.50       326 - 500
5       5.00      0.12      0.62       501 - 625
6       6.00      0.12      0.75       626 - 750
7       7.00      0.12      0.88       751 - 875
8       8.00      0.12      1.00       876 - 1000

-- TICKET TABLE --
------------------
n      PRICE     PROB.      CDF         RANGE
1       1.00      1.00      1.00         1 -  1
```

Diagram 4.6 Display the Service table, Interarrival table and Ticket table



```
START SIMULATION

[ 0 ] Customer 1 has arrived.
[ 0 ] Customer 1 has entered Counter 1.

[ 1 ] Counter 1 has finished serving Customer 1.
[ 1 ] Customer 2 has arrived.
[ 1 ] Customer 2 has entered Counter 1.


[ 3 ] Customer 3 has arrived.
[ 3 ] Customer 3 has entered Counter 1.

[ 4 ] Counter 1 has finished serving Customer 2.
[ 4 ] Counter 1 has started serving Customer 3.



[ 7 ] Counter 1 has finished serving Customer 3.

[ 8 ] Customer 4 has arrived.
[ 8 ] Customer 4 has entered Counter 1.




[ 12 ] Customer 5 has arrived.
[ 12 ] Customer 5 has entered Counter 1.


[ 14 ] Counter 1 has finished serving Customer 4.
[ 14 ] Counter 1 has started serving Customer 5.



[ 17 ] Customer 6 has arrived.
[ 17 ] Customer 6 has entered Counter 1.


[ 19 ] Counter 1 has finished serving Customer 5.
```

Diagram 4.7 Display the step-by-step simulation



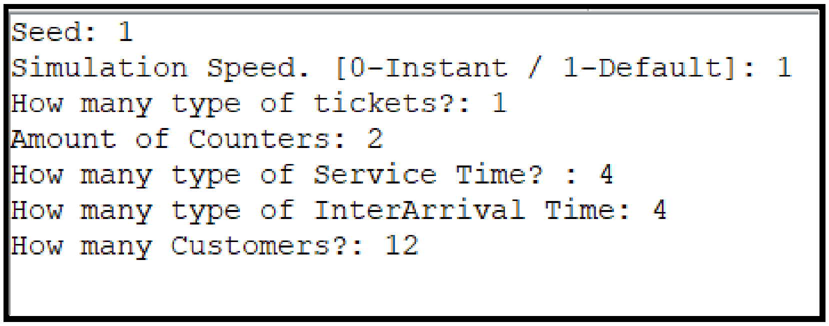
```
 ----------------------
 -- CUSTOMER SUMMARY --
 ----------------------
 n  RN Inter    Inter.   Arrival    RN Tkt  Tkt Type   Tkt Qty  Amt. Paid
 1         8         1         0         1         1         1         1
 2        36         1         1         1         1         1         1
 3       148         2         3         1         1         1         1
 4       596         5         8         1         1         1         1
 5       388         4        12         1         1         1         1
 6       556         5        17         1         1         1         1
 7       228         2        19         1         1         1         1
 8       916         8        27         1         1         1         1
 9       668         6        33         1         1         1         1
10       676         6        39         1         1         1         1


 ---------------------
 -- COUNTER SUMMARY --
 ---------------------
 Counter 1
 -------------
 n   RN Serv   Serv.  Serv Start  Serv End   Waiting  Time in System
 1         8       1           0         1         0         1
 2        36       3           1         4         0         3
 3        48       3           4         7         1         4
 4        96       6           8        14         0         6
 5        88       5          14        19         2         7
 6        56       3          19        22         2         5
 7        28       2          22        24         3         5
 8        16       2          27        29         0         2
 9        68       4          33        37         0         4
10        76       4          39        43         0         4


 ------------------------
 -- SIMULATION SUMMARY --
 ------------------------
Average Waiting Time: 0.80
Average InterArrival Time Time: 4.00
Average Arrival Time: 15.90
Average Time Spent in System: 4.10

Counter 1 Average Service: 3.30
Counter 1 Total Sales: 10.00
```

Diagram 4.8 Display the summary for customer table, counter table and the simulation average value

Input and Output of 2 Counter

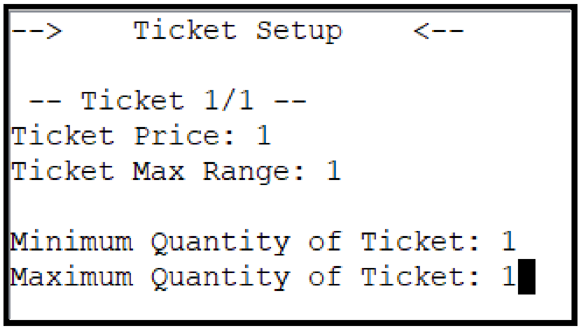
For this simulation we have decided to use the value of 1 as the seed number. Besides that, for random number generators we used mixed LCG with the value for ‘a’ is 4 and the value of ‘c’ is 3.

Input :



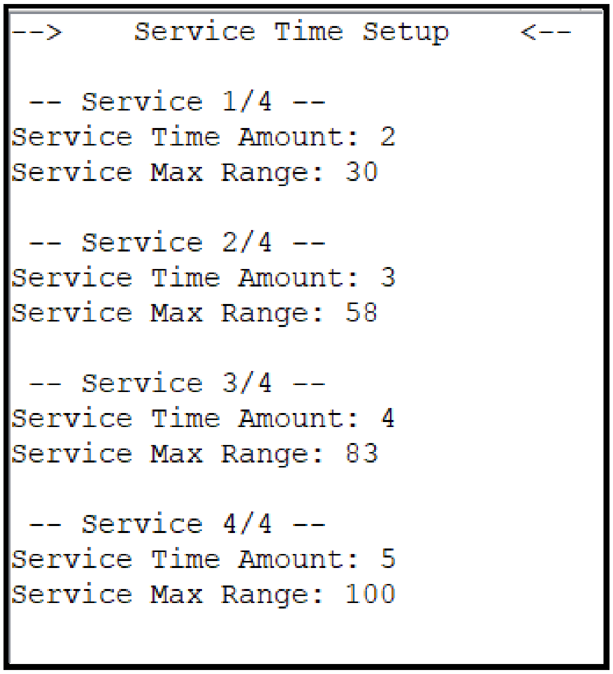
```
Seed: 1
Simulation Speed. [0-Instant / 1-Default]: 1
How many type of tickets?: 1
Amount of Counters: 2
How many type of Service Time? : 4
How many type of InterArrival Time: 4
How many Customers?: 12
```

Diagram 4.9 Initialize the variable needed for simulation



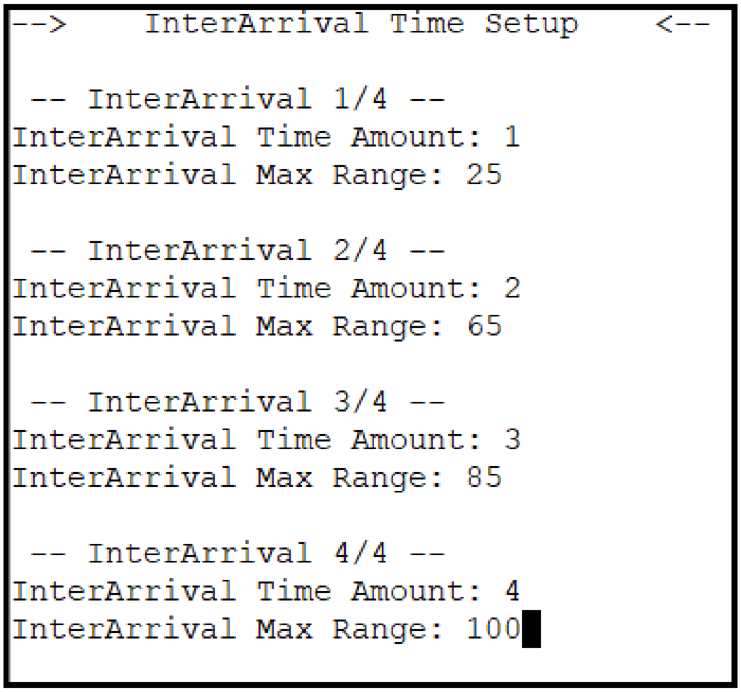
```
-->     Ticket Setup    <--

 -- Ticket 1/1 --
Ticket Price: 1
Ticket Max Range: 1

Minimum Quantity of Ticket: 1
Maximum Quantity of Ticket: 1
```

Diagram 4.10 Initialize the price and quantity of ticket



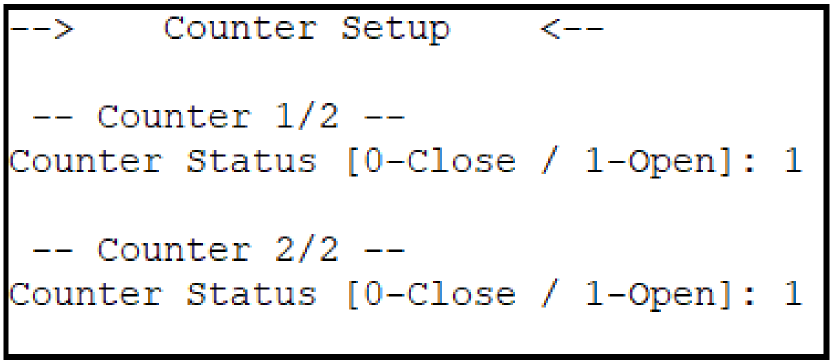
```
-->     Service Time Setup    <--

 -- Service 1/4 --
Service Time Amount: 2
Service Max Range: 30

 -- Service 2/4 --
Service Time Amount: 3
Service Max Range: 58

 -- Service 3/4 --
Service Time Amount: 4
Service Max Range: 83

 -- Service 4/4 --
Service Time Amount: 5
Service Max Range: 100
```

Diagram 4.11 Initialize the value for service time table



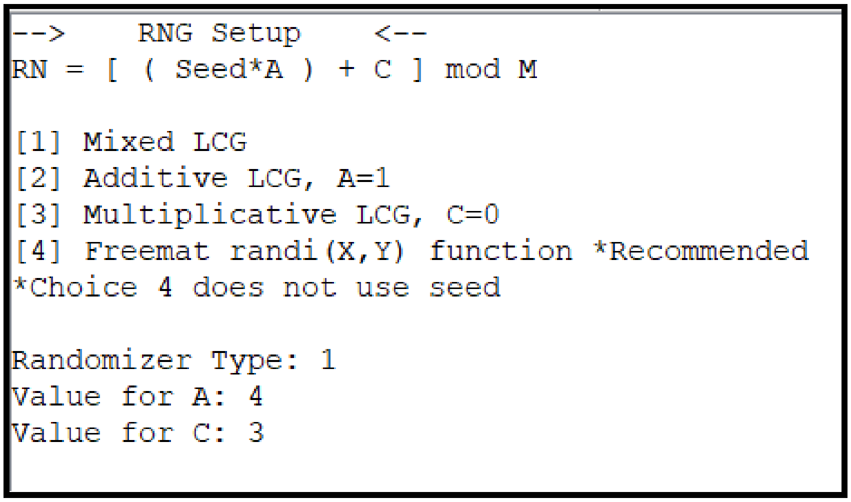
```
-->     InterArrival Time Setup    <--

 -- InterArrival 1/4 --
InterArrival Time Amount: 1
InterArrival Max Range: 25

 -- InterArrival 2/4 --
InterArrival Time Amount: 2
InterArrival Max Range: 65

 -- InterArrival 3/4 --
InterArrival Time Amount: 3
InterArrival Max Range: 85

 -- InterArrival 4/4 --
InterArrival Time Amount: 4
InterArrival Max Range: 100
```

Diagram 4.12 Initialize the value for interarrival time table



```
-->     Counter Setup     <--

 -- Counter 1/2 --
Counter Status [0-Close / 1-Open]: 1

 -- Counter 2/2 --
Counter Status [0-Close / 1-Open]: 1
```

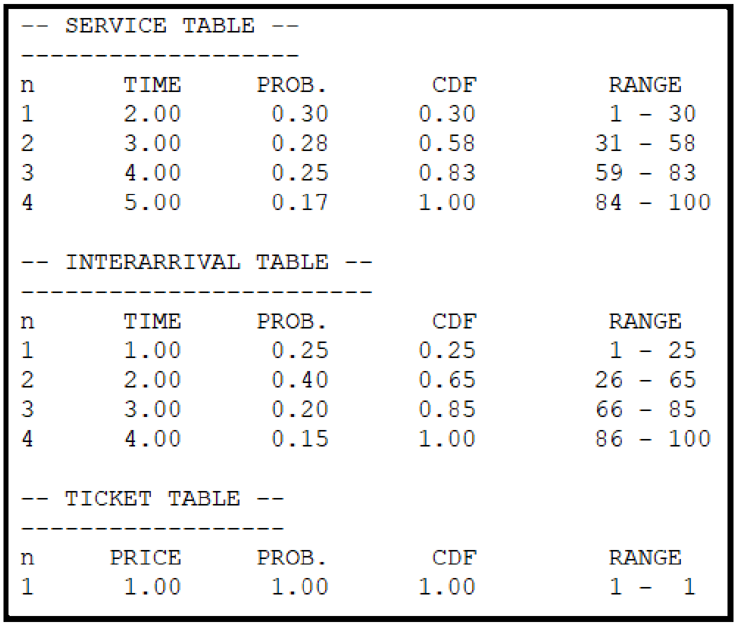
Diagram 4.13 Choose the counter status



```
-->     RNG Setup     <--
RN = [ ( Seed*A ) + C ] mod M

[1] Mixed LCG
[2] Additive LCG, A=1
[3] Multiplicative LCG, C=0
[4] Freemat randi(X,Y) function *Recommended
*Choice 4 does not use seed

Randomizer Type: 1
Value for A: 4
Value for C: 3
```

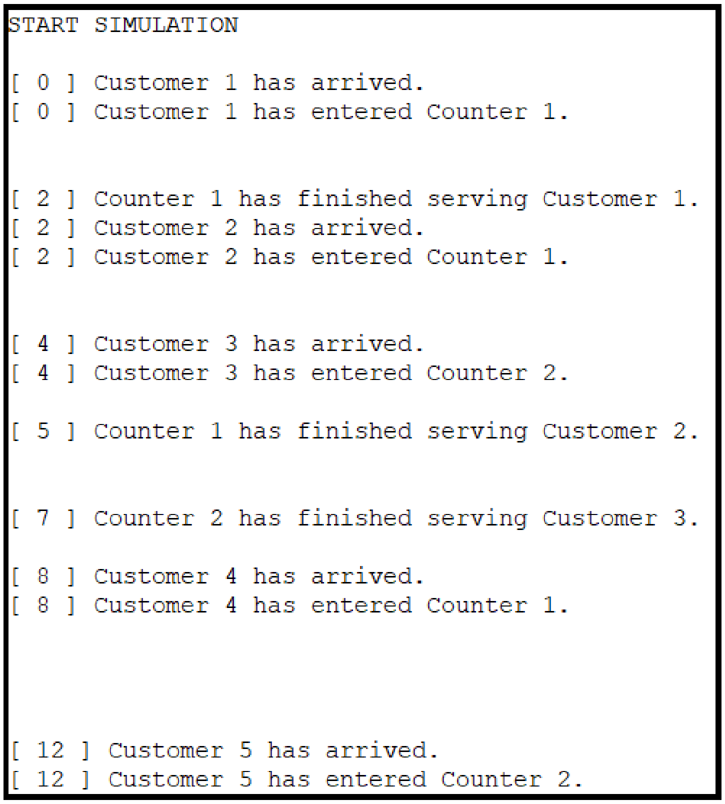
Diagram 4.14 Choose the Random Number Generator Setup

Output :



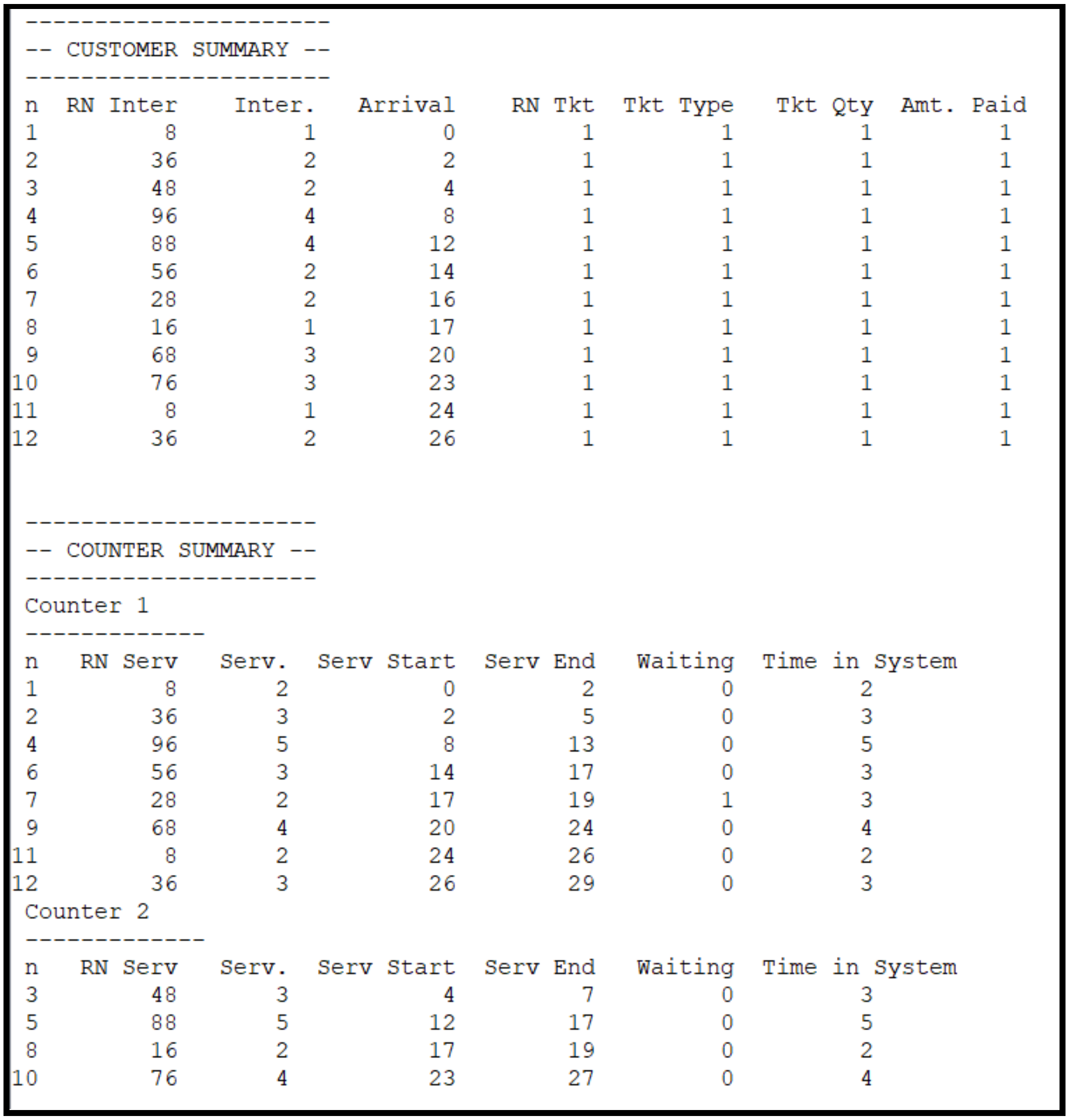
```
-- SERVICE TABLE --
-------------------
n      TIME     PROB.      CDF         RANGE
1      2.00      0.30     0.30          1 - 30
2      3.00      0.28     0.58         31 - 58
3      4.00      0.25     0.83         59 - 83
4      5.00      0.17     1.00         84 - 100

-- INTERARRIVAL TABLE --
------------------------
n      TIME     PROB.      CDF         RANGE
1      1.00      0.25     0.25          1 - 25
2      2.00      0.40     0.65         26 - 65
3      3.00      0.20     0.85         66 - 85
4      4.00      0.15     1.00         86 - 100

-- TICKET TABLE --
------------------
n     PRICE     PROB.      CDF         RANGE
1      1.00      1.00     1.00          1 -  1
```

Diagram 4.15 Display the Service table, Interarrival table and Ticket table



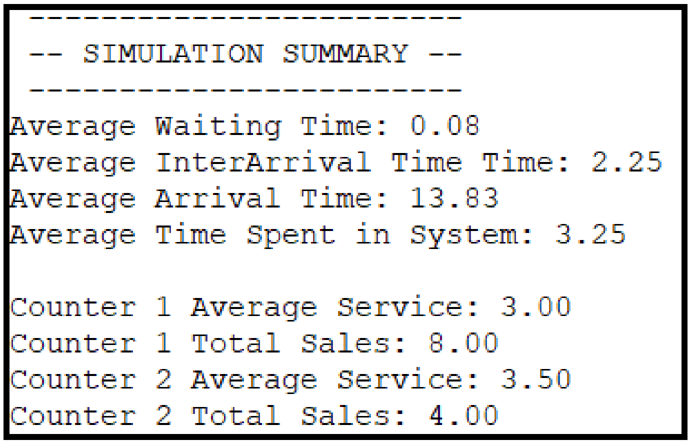
```
START SIMULATION

[ 0 ] Customer 1 has arrived.
[ 0 ] Customer 1 has entered Counter 1.


[ 2 ] Counter 1 has finished serving Customer 1.
[ 2 ] Customer 2 has arrived.
[ 2 ] Customer 2 has entered Counter 1.


[ 4 ] Customer 3 has arrived.
[ 4 ] Customer 3 has entered Counter 2.

[ 5 ] Counter 1 has finished serving Customer 2.


[ 7 ] Counter 2 has finished serving Customer 3.

[ 8 ] Customer 4 has arrived.
[ 8 ] Customer 4 has entered Counter 1.




[ 12 ] Customer 5 has arrived.
[ 12 ] Customer 5 has entered Counter 2.
```

Diagram 4.16 Display the step-by-step simulation



```
 ----------------------
 -- CUSTOMER SUMMARY --
 ----------------------
 n  RN Inter     Inter.   Arrival    RN Tkt  Tkt Type   Tkt Qty  Amt. Paid
 1         8          1         0         1         1         1         1
 2        36          2         2         1         1         1         1
 3        48          2         4         1         1         1         1
 4        96          4         8         1         1         1         1
 5        88          4        12         1         1         1         1
 6        56          2        14         1         1         1         1
 7        28          2        16         1         1         1         1
 8        16          1        17         1         1         1         1
 9        68          3        20         1         1         1         1
10        76          3        23         1         1         1         1
11         8          1        24         1         1         1         1
12        36          2        26         1         1         1         1


 ---------------------
 -- COUNTER SUMMARY --
 ---------------------
 Counter 1
 -------------
 n   RN Serv   Serv.  Serv Start  Serv End   Waiting  Time in System
 1         8       2          0          2         0         2
 2        36       3          2          5         0         3
 4        96       5          8         13         0         5
 6        56       3         14         17         0         3
 7        28       2         17         19         1         3
 9        68       4         20         24         0         4
11         8       2         24         26         0         2
12        36       3         26         29         0         3
 Counter 2
 -------------
 n   RN Serv   Serv.  Serv Start  Serv End   Waiting  Time in System
 3        48       3          4          7         0         3
 5        88       5         12         17         0         5
 8        16       2         17         19         0         2
10        76       4         23         27         0         4
```

Diagram 4.17 Display the summary for customer table and both counter table



```
 ------------------------
 -- SIMULATION SUMMARY --
 ------------------------
Average Waiting Time: 0.08
Average InterArrival Time Time: 2.25
Average Arrival Time: 13.83
Average Time Spent in System: 3.25

Counter 1 Average Service: 3.00
Counter 1 Total Sales: 8.00
Counter 2 Average Service: 3.50
Counter 2 Total Sales: 4.00
```

Diagram 4.18 The summary of the simulation