Faculty of Computing and Informatics (FCI)

Multimedia University, Cyberjaya

TMA1301 Computational Methods

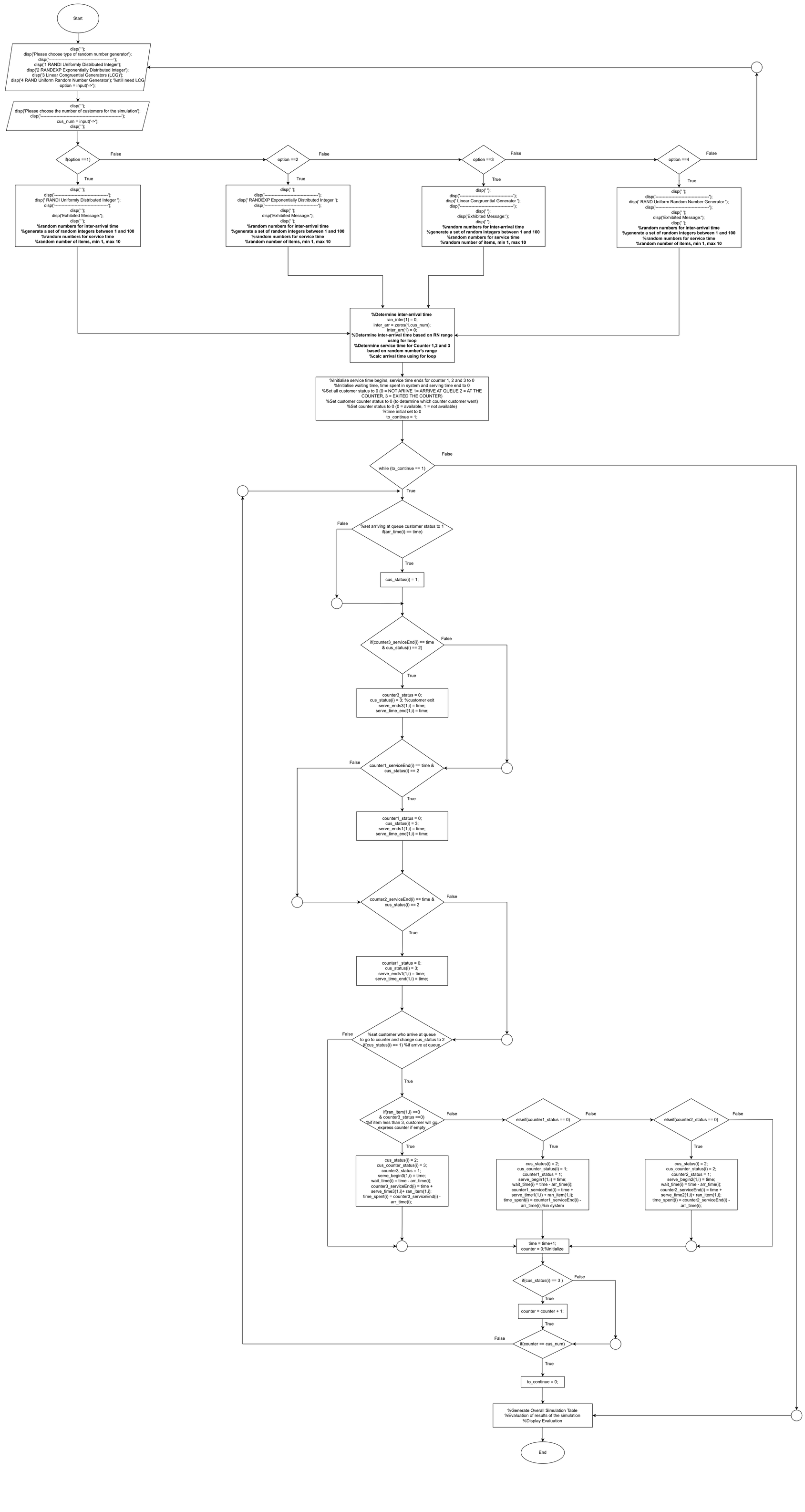
Tri 2, 2022/2023

Coding Assignment Report

Tutorial Section: TT7L

Group 3

Name	Student ID
EMILY PHANG RU YING	1211102687
LIM CAI QING	1211102753
TEO YU JIE	1211102751
HO JING LU	1221302735
CHAN YENG HUI	1221303663


Start
disp(' ');
disp('Please choose type of random number generator');
disp('----------');
disp('1 RANDI Uniformly Distributed Integer');
disp('2 RANDEXP Exponentially Distributed Integer');
disp('3 Linear Congruential Generators (LCG)');
disp('4 RAND Uniform Random Number Generator'); %still need LCG
option = input('->');
disp(' ');
disp('Please choose the number of customers for the simulation');
disp('----------');
cus_num = input('->');
disp(' ');
if(option ==1)
False
True
option ==2
False
True
option ==3
False
True
option ==4
False
True
disp(' ');
disp('----------');
disp(' RANDI Uniformly Distributed Integer ');
disp('----------');
disp(' ');
disp('Exhibited Message:');
disp(' ');
%random numbers for inter-arrival time
%generate a set of random integers between 1 and 100
%random numbers for service time
%random number of items, min 1, max 10
disp(' ');
disp('----------');
disp(' RANDEXP Exponentially Distributed Integer ');
disp('----------');
disp(' ');
disp('Exhibited Message:');
disp(' ');
%random numbers for inter-arrival time
%generate a set of random integers between 1 and 100
%random numbers for service time
%random number of items, min 1, max 10
disp(' ');
disp('----------');
disp(' Linear Congruential Generator ');
disp('----------');
disp(' ');
disp('Exhibited Message:');
disp(' ');
%random numbers for inter-arrival time
%generate a set of random integers between 1 and 100
%random numbers for service time
%random number of items, min 1, max 10
disp(' ');
disp('----------');
disp(' RAND Uniform Random Number Generator ');
disp('----------');
disp(' ');
disp('Exhibited Message:');
disp(' ');
%random numbers for inter-arrival time
%generate a set of random integers between 1 and 100
%random numbers for service time
%random number of items, min 1, max 10
%Determine inter-arrival time
ran_inter(1) = 0;
inter_arr = zeros(1,cus_num);
inter_arr(1) = 0;
%Determine inter-arrival time based on RN range
using for loop
%Determine service time for Counter 1,2 and 3
based on random number's range
%calc arrival time using for loop
%Initialise service time begins, service time ends for counter 1, 2 and 3 to 0
%Initialise waiting time, time spent in system and serving time end to 0
%Set all customer status to 0 (0 = NOT ARIIVE 1= ARRIVE AT QUEUE 2 = AT THE COUNTER, 3 = EXITED THE COUNTER)
%Set customer counter status to 0 (to determine which counter customer went)
%Set counter status to 0 (0 = available, 1 = not available)
%time initial set to 0
to_continue = 1;
while (to_continue == 1)
False
True
%set arriving at queue customer status to 1
if(arr_time(i) == time)
False
True
cus_status(i) = 1;
if(counter3_serviceEnd(i) == time
& cus_status(i) == 2)
False
True
counter3_status = 0;
cus_status(i) = 3; %customer exit
serve_ends3(1,i) = time;
serve_time_end(1,i) = time;
counter1_serviceEnd(i) == time &
cus_status(i) == 2
False
True
counter1_status = 0;
cus_status(i) = 3;
serve_ends1(1,i) = time;
serve_time_end(1,i) = time;
counter2_serviceEnd(i) == time &
cus_status(i) == 2
False
True
counter1_status = 0;
cus_status(i) = 3;
serve_ends1(1,i) = time;
serve_time_end(1,i) = time;
%set customer who arrive at queue
to go to counter and change cus_status to 2
if(cus_status(i) == 1) %if arrive at queue
False
True
if(ran_item(1,i) <=3
& counter3_status ==0)
%if item less than 3, customer will go
express counter if empty
False
True
elseif(counter1_status == 0)
False
True
elseif(counter2_status == 0)
False
True
cus_status(i) = 2;
cus_counter_status(i) = 3;
counter3_status = 1;
serve_begin3(1,i) = time;
wait_time(i) = time - arr_time(i);
counter3_serviceEnd(i) = time +
serve_time3(1,i)+ ran_item(1,i);
time_spent(i) = counter3_serviceEnd(i) -
arr_time(i);
cus_status(i) = 2;
cus_counter_status(i) = 1;
counter1_status = 1;
serve_begin1(1,i) = time;
wait_time(i) = time - arr_time(i);
counter1_serviceEnd(i) = time +
serve_time1(1,i) + ran_item(1,i);
time_spent(i) = counter1_serviceEnd(i) -
arr_time(i);%in system
cus_status(i) = 2;
cus_counter_status(i) = 2;
counter2_status = 1;
serve_begin2(1,i) = time;
wait_time(i) = time - arr_time(i);
counter2_serviceEnd(i) = time +
serve_time2(1,i)+ ran_item(1,i);
time_spent(i) = counter2_serviceEnd(i) -
arr_time(i);
time = time+1;
counter = 0;%initialize
if(cus_status(i) == 3)
False
True
counter = counter + 1;
True
if(counter == cus_num)
False
True
to_continue = 0;
%Generate Overall Simulation Table
%Evaluation of results of the simulation
%Display Evaluation
End

Introduction & Description of Simulation

The simulation presented in this report is an imitation of real-life queuing for counter checkouts in supermarkets. By using random number generators, namely Uniformly Distributed Integer, Exponentially Distributed Integer, Linear Congruential Generator and Uniform Random Number Generator, real-life situation is imitated to predict a number of operations, such as number of items acquired by customers, customers' inter-arrival time and the service time of a particular counter.

Users are given the option to choose the random number generator desired, and the number of customers to be input. The program will receive user's input and generate the probability tables of the counters and inter-arrival time.

The probability tables of the counters and inter-arrival time are subsequently displayed, each containing service time or inter-arrival time depending on the type of table, followed by the probability, cumulative distribution function(CDF) and the random number range that corresponds with the probability that is output.

The generator chosen by the user will be used to generate the simulation table. The simulation tables of the inter-arrival time and the counters are then displayed.

The simulation table of inter-arrivals displays the random number generated from the random number generator, inter-arrival times of customers that corresponds to the random number, the arrival times of customers calculated from the inter-arrival times and the number of items acquired by the customers.

The simulation tables of the counters shows the generator generated random number for service time, where the service time column will refer to the probability table that corresponds to the random number range to output the service time.

Time spent in the system is dependent on the number of items a customer has, time of service beginning and the service time. Customers with less than 4 items are more likely to queue up in the express counter.

Messages are exhibited every time a customers enter and leaves a system, and the average calculation for entries of the simulation tables are calculated in the end.

Random Number Generator

RANDI Uniformly Distributed Integer

```
if(option ==1)
    disp(' ');
    disp('-------------------------------------');
    disp(' RANDI Uniformly Distributed Integer ');
    disp('-------------------------------------');
    disp(' ');
    disp('Exhibited Message:');
    disp(' ');

    %generate inter-arrival RN
    %random numbers for inter-arrival time
    %generate a set of random integers between 1 and 100
    ran_inter = [randi(ones(1,cus_num), 100*ones(1,cus_num))];


    %random numbers for service time
    ran_serve = [randi(ones(1,cus_num), 100*ones(1,cus_num))];



    %random number of items, min 1, max 10
    ran_item = zeros(1,cus_num);
    for(i = 1:cus_num)

        ran_item(i) = randi(1, 10);
    end
```

When the user chooses option 1, the code generates random numbers for inter-arrival time, service time, and the number of items using RANDI Uniformly Distributed Integer. The inter-arrival times and service times are generated using the randi function, while the number of items is generated using the randi function within a for loop for the total number of customers. Randi is a built-in freemat function that generate an array of uniformly distributed integers between the two supplied limits. For random numbers for inter-arrival time, service time, we used the randi function to generate a set of random integers between 1 and 100. The ones(1,cus_num) creates a vector of ones with a length of cus_num, and it is multiplied by 100 using 100*ones(1,cus_num) to set the upper limit for the random integers. The resulting random inter-arrival times and random service times are stored in the variable ran_inter and ran_serve. Within the loop, the line ran_item(i) = randi(1, 10) generates a random number between 1 and 10 using the randi function. The resulting random number is assigned to the i-th element of the ran_item vector.

RANDEXP Exponentially Distributed Integer

```
elseif(option==2)
    disp(' ');
    disp('-----------------------------------------');
    disp(' RANDEXP Exponentially Distributed Integer');
    disp('-----------------------------------------');
    disp(' ');
    disp('Exhibited Message:');
    disp(' ');

    %random numbers for inter-arrival time
    ran_inter = zeros(1,cus_num);

 for i = 1:cus_num

     lamda=(100-1)/2;
     ran_inter(i)=round(randexp(lamda))+1;

    while ran_inter(i)>100
        ran_inter(i)=round(randexp(lamda))+1;
    end

 end

%random numbers for service time
ran_serve = zeros(1,cus_num);

for i = 1:cus_num

    lamda=(100-1)/2;
    ran_serve(i)=round(randexp(lamda))+1;

    while ran_serve(i)>100
        ran_serve(i)=round(randexp(lamda))+1;
    end

end


%random number of items, min 1, max 10
ran_item = zeros(1,cus_num);
    for(i = 1:cus_num)

        ran_item(i) = round(randexp((10-1)/2))+1;

        while ran_item(i)>10
        ran_item(i) = round(randexp((10-1)/2))+1;
        end

    end
```

When the user chooses option 2, the randexp function will be used to generate Exponentially Distributed Integers for the service time and interarrival time.

Generating random numbers for interarrival time

The code generates ‘cus_num’ random numbers. ‘cus_num’ stores the user input of number of customers. The random numbers generated are stored in ‘ran_inter’.

The ‘for’ loop iterates as many times as the value of ‘cus_num’. Inside the loop, ‘lamda’ is set at (100-1)/2 which is the mean occurrence per unit time. The random number generated by the randexp function is then rounded. 1 is added to the rounded value to ensure the number generated is larger than 0. The value is then stored in ‘ran_inter’. A ‘while’ loop is added to check if the generated value exceeds 100, then regenerate if it does until the number generated is equal or less than 100.

The 1st index of ‘ran_inter’ will be set to 0 later as first customer do not have interarrival time.

Generating random numbers for service time

The random numbers for service time is generated in a similar manner as interarrival time. However, the numbers generated will be stored in ‘ran_serve’ instead of ‘ran_inter’.

Generating random numbers for number of items

The code generates ‘cus_num’ random numbers. ‘cus_num’ stores the user input of number of customers. The random numbers generated are stored in ‘ran_item’. The minimum number of item is 1 and the maximum is 10.

The ‘for’ loop iterates as many times as the value of ‘cus_num’. Inside the loop, the mean occurrence is set at (10-1)/2. The random number generated by the randexp function is then rounded. 1 is added to the rounded value to ensure the number generated is larger than 0. The value is then stored in ‘ran_item’. A ‘while’ loop is added to check if the generated value exceeds 10, then regenerate if it does until the number generated is equal or less than 10.

Linear Congruential Generator

```
elseif (option == 3)
    disp(' ');
    disp('-------------------------------------');
    disp('    Linear Congruential Generator    ');
    disp('-------------------------------------');
    disp(' ');
    disp('Exhibited Message:');
    disp(' ');

    %random numbers for inter-arrival time
    %generate a set of random integers between 1 and 100
    ran_inter = round(mod(rand(1, cus_num) * 100, 99)) + 1;


    %random numbers for service time
    ran_serve = round(mod(rand(1, cus_num) * 100, 99)) + 1;


    %random number of items, min 1, max 10
    ran_item = zeros(1,cus_num);
    for(i = 1:cus_num)

        ran_item(i) =  round(mod(rand * 10,9)) + 1;
    end
```

When the user chooses option 3, the code above outputs a random number based on the Linear Congruential Generator. There is no built-in function available for Linear Congruential Generator in FreeMat therefore other built-in functions are utilized to form the Linear Congruential Generator.

rand(1, cus_num) * 100 would display random numbers within 100 in (1 × cus_num) matrix. mod(rand(1, cus_num) * 100, 99) + 1 shifts the range so there would be no 0 value and adding the built-in round(mod(rand(1,cus_num) * 100, 99)) + 1 rounds the number to an integer..

In ran_item, the variable is first initialised with 0 with a (1 × cus_num) matrix so values can be inserted. A for loop is used to interate from i =1 to i = cus_num. In the loop, round(mod(rand * 10, 9)) + 1 generates random numbers between 1 and 10 and store the number in ran_item(i).

RAND Uniform Random Number Generator

```
elseif(option ==4)
    disp(' ');
    disp('-------------------------------------');
    disp(' RAND Uniform Random Number Generator ');
    disp('-------------------------------------');
    disp(' ');
    disp('Exhibited Message:');
    disp(' ');

    %random numbers for inter-arrival time
    %generate a set of random integers between 1 and 100
    for(i = 1:cus_num)
        A = 1; B = 100;
        ran_inter = [round(A + (B-A)*rand(1,i))];
    end

    %random numbers for service time
    for(i = 1:cus_num)
        A = 1; B = 100;
        ran_serve = [round(A + (B-A)*rand(1,i))];
    end

    %random number of items, min 1, max 10
    ran_item = zeros(1,cus_num);
    for(i = 1:cus_num)

        ran_item(i) = randi(1, 10);
    end

end
```

When the user chooses option 4, the code generates random numbers for inter-arrival time, service time, items (min 1, max 10) using RAND Uniform Random Number Generator. This is a loop that starts with i being 1 and iterates up to cus_num. In each iteration, the code within the loop is executed. Inside the loop, two variables, A and B are defined with the values 1 and 100. These variables represent the lower and upper bounds of the desired range for the random numbers.The line 'ran_inter = [round(A + (B-A)*rand(1, i))];' generates a random array of integers with a size determined by the loop variable i. In the first iteration, i is 1, so a single random integer is generated. In the second iteration, i is 2, so a random array of 2 integers is generated, and so on. The generated random numbers are scaled to fit within the range specified by A and B, and then rounded to the nearest integer (also same as service time). The loop will continue to iterate cus_num times, and with each iteration, a new random array is generated and any subsequent operations are performed.

Generation of Table for Service Time for 3 Servers & Inter-Arrival Time

Functions

Function probability

```
function output=probability
    cdf=0;

    div=1/(6);
    offset=div;

    upper=div+offset;
    lower=1/5*0.65;

    for i=1:4
        P(i)=round((rand()*(upper-lower)+lower)*100)/100;

        if (P(i) > div)
            offset=offset-(P(i)-div);
            upper=div+offset;
        end

        cdf=cdf+P(i);
    end

    P(5)=1-cdf;

    output=P;
```

The function generates 5 random probability distribution values which adds up to a cdf of 1. It returns the output ‘P’ which is a 1x5 matrix which store the random probability values generated.

The variables are initialised as below:

-‘div’ is the division of 1 by 6, which represents a fraction.

-‘offset’ is initially set as ‘div’.

-‘upper’ is set as ‘div + offset’.

-‘lower’ is set as ‘(1/5) * 0.65’, which is a specific value used as the lower limit of the random values to be generated.

A ‘for’ loop reiterates 4 times to generate random values of probability to be stored in the first 4 index of the ‘P’. Inside the loop, a random number between 0 and 1 is generated. It is then multiplied by the range (upper - lower) and add lower to ensure it falls within the desired range. The number is then multiplied by 100, rounded, then divide by 100 again. The calculated value is then stored in ‘P’.

The generated value ‘P(i)’ is checked if it is greater than ‘div’. If true, calculate the difference ‘(P(i) - div)’ and subtract it from the ‘offset’. Then, update the ‘upper’ as ‘div + offset’.

The generated value ‘P(i)’ is added to the cumulative distribution function ‘cdf’.

‘P(5)’ is calculated by subtracting ‘cdf’ obtain in the loop from 1.

Function CDF

```
function output = CDF(prob)
    S = zeros(1,5);
    cdf = 0;

    for i = 1:length(prob)
        cdf = cdf + prob(i);
        S(i) = cdf;
    end
    output = S;
end
```

The function calculates the cumulative distribution function (CDF). It accepts an input ‘prob’, a 1x5 matrix which stores the 5 values of probability distribution generated beforehand. It returns an output ‘S’, a 1x5 matrix which stores the cdf for each element of the input ‘prob’.

The loop reiterates 5 times, which is the length of the input ‘prob’ matrix. Inside the loop, the current probability ‘prob(i)’ is added to the cumulative sum ‘cdf’. The cumulative sum ‘cdf’ is then assigned to ‘S’ at index ‘i’. This stores the cumulative probability up to that point.

Function LowRange

```
function output = LowRange(cdf)
    L = zeros(1,5);
    L(1) = 1;
    Lowrange = 0;

    for i = 2:5
        Lowrange = round(cdf(i-1)*100+1);
        L(i) = Lowrange;
    end
    output = L;
end
```

The function calculates the lower limit of the random number range. It accepts an input ‘cdf’, a 1x5 matrix which stores the 5 values of cdf calculated beforehand. It returns an output ‘L’, a 1x5 matrix which stores the lower limit of the random number range for each probability distribution.

First index of ‘L’ is set a 1. The loop reiterates 4 times, to calculate the values assigned to ‘L’ from index 2 to 5. Inside the loop, the previous element of the CDF ‘cdf(i-1)’ is multiplied by 100, adding 1, and then rounded. The calculated value is the lower limit of the random number range corresponding to the current probability distribution. The value is then assigned to ‘L’ at index ‘i’.

Function UppRange

```
function output = UppRange(cdf)
    U = zeros(1,5);
    U(5) = 100;
    Upprange = 0;

    for i = 1:4
        Upprange = round(cdf(i)*100);
        U(i) = Upprange;
    end
    output = U;
end
```

The function calculates the upper limit of the random number range. It accepts an input ‘cdf’, a 1x5 matrix which stores the 5 values of cdf calculated beforehand. It returns an output ‘U’, a 1x5 matrix which stores the upper limit of the random number range for each probability distribution.

The loop reiterates 4 times, to calculate the values assigned to ‘U’ from index 1 to 4. Inside the loop, the value of ‘cdf’ at index ‘i’ is multiplied by 100, then rounded. The calculated value is the upper limit of the random number range corresponding to the current probability distribution. The value is then assigned to ‘U’ at index ‘i’. Last index of ‘U’ is set at 100.

Counter 1 service time

```
%counter 1 service time
service1_time = 3:7;
service1_prob = probability();
service1_cdf = CDF(service1_prob);
service1_Lowrange = LowRange(service1_cdf);
service1_Upprange = UppRange(service1_cdf);
disp(' ');
    disp('COUNTER 1 SERVICE TIME PROBABILITY TABLE');
    disp('-------------------------------------------------------------------------');
    disp('Service time for Counter 1(min)| Probability | CDF  | Random number range');
        for (i=1:5)
            fprintf('              %1.0f              |    %1.2f     | %1.2f |      %2.0f-%2.0f   \n',
        end
```

The code generates the probability table of counter 1 service time.

'service1_time' stores the service time for counter 1 which ranges from 3 to 7 minutes.

Probability distribution for counter 1 service time is calculated by calling the 'probability' function and inputting 'service1_time' to the function. The output is then stored in 'service1_prob'.

CDF for counter 1 is calculated by calling the 'CDF' function and inputting 'service1_prob' to the function. The output is then stored in 'service1_cdf'.

The lower limit of the random number range of counter 1 is calculated by calling the 'LowRange' function and inputting 'service1_cdf' to the function. The output is then stored in 'service1_Lowrange'.

The upper limit of the random number range of counter 1 is calculated by calling the 'UppRange' function and inputting 'service1_cdf' to the function. The output is then stored in 'service1_Upprange'.

'service1_Lowrange' and 'service1_Upprange' together forms the random number range for each service time in counter 1.

The table is then generated by using a 'for' loop which iterates through each index of the matrix mentioned above to display service time for Counter 1 in min, Probability, CDF and Random number range.

Counter 2, Counter 3 service time and Interarrival time

```
%counter 2 service time
service2_time = 3:7;
service2_prob = probability();
service2_cdf = CDF(service2_prob);
service2_Lowrange = LowRange(service2_cdf);
service2_Upprange = UppRange(service2_cdf);
disp(' ');
    disp('COUNTER 2 SERVICE TIME PROBABILITY TABLE');
    disp('-------------------------------------------------------------------------');
    disp('Service time for Counter 2(min)| Probability | CDF  | Random number range');
        for (i=1:5)
            fprintf('               %1.0f               |    %1.2f     | %1.2f |      %2.0f-%2.0f   \n',
        end
```

```
%counter 3 service time (express counter)
service3_time = 1:5;
service3_prob = probability();
service3_cdf = CDF(service3_prob);
service3_Lowrange = LowRange(service3_cdf);
service3_Upprange = UppRange(service3_cdf);
disp(' ');
    disp('COUNTER 3 SERVICE TIME PROBABILITY TABLE (EXPRESS COUNTER)');
    disp('-------------------------------------------------------------------------');
    disp('Service time for Counter 3(min)| Probability | CDF  | Random number range');
        for (i=1:5)
            fprintf('               %1.0f               |    %1.2f     | %1.2f |      %2.0f-%2.0f   \n',
        end
```

```
%interarrival time
inter_arr_time = 1:5;
inter_arr_prob = probability();
inter_arr_cdf = CDF(inter_arr_prob);
inter_arr_Lowrange = LowRange(inter_arr_cdf);
inter_arr_Upprange = UppRange(inter_arr_cdf);
disp(' ');
    disp('INTER ARRIVAL TIME PROBABILITY TABLE');
    disp('-------------------------------------------------------------------------');
    disp('Interarrival time(min)| Probability | CDF  | Random number range');
        for (i=1:5)
            fprintf('           %1.0f           |    %1.2f     | %1.2f |      %2.0f-%2.0f   \n', [i
        end
```

The probability table of service time for counter 2, counter 3 and customer interarrival time is generated in a similar manner as counter 1. However, the service times of counter 1, 2 and 3 varies. Service time of counter 2 ranges from 3 to 7 minutes whereas service time of counter 3 ranges from 1 to 5 minutes as it is an express counter. The interarrival time ranges from 1 to 5 minutes.

Determine Inter-arrival and Sevice Time based on Random Number's range

```
%Determine inter-arrival time

ran_inter(1) = 0;   % to set the first index as 0

    %inter arrival time based on random number's range
    inter_arr = zeros(1,cus_num);
    inter_arr(1) = 0;   % to set the first index as 0

    for(i = 2:cus_num)
        if(ran_inter(1,i) >= inter_arr_Lowrange(1,1) & ran_inter(1,i) <= inter_arr_Upprange(1,1))
            inter_arr(i) = inter_arr_time(1,1);
        elseif(ran_inter(1,i) >= inter_arr_Lowrange(1,2) & ran_inter(1,i) <= inter_arr_Upprange(1,2))
            inter_arr(i) = inter_arr_time(1,2);
        elseif(ran_inter(1,i) >= inter_arr_Lowrange(1,3) & ran_inter(1,i) <= inter_arr_Upprange(1,3))
            inter_arr(i) = inter_arr_time(1,3);
        elseif(ran_inter(1,i) >= inter_arr_Lowrange(1,4) & ran_inter(1,i) <= inter_arr_Upprange(1,4))
            inter_arr(i) = inter_arr_time(1,4);
        elseif(ran_inter(1,i) >= inter_arr_Lowrange(1,5) & ran_inter(1,i) <= inter_arr_Upprange(1,5))
            inter_arr(i) = inter_arr_time(1,5);
        end
    end
```

The first index of random inter-arrival time is set as 0 as customer 1 does not need to have RN for inter-arrival time.The inter arrival time array is initialized with zeros and has the same size as the number of customers (cus_num). The first index is set to 0. The code then iterates over each customer using the variable i, starting from the second customer (i = 2). For each customer, the code checks the value of inter-arrival's RN (ran_inter(1, i)) to determine which range it falls into. The ranges are defined by inter_arr_Lowrange and inter_arr_Upprange arrays. If inter-arrival's RN falls within the range specified by inter_arr_Lowrange(1, 1) and inter_arr_Upprange(1, 1), the inter-arrival time for that customer is set to inter-arrival time (inter_arr_time(1, 1)). The same process is repeated for the other ranges using the appropriate indices. The resulting inter-arrival times are then stored in the inter_arr array.

```
%service time Counter 1 based on random number's range
    serve_time1 = zeros(1,cus_num);
    for(i = 1:cus_num)
        if(ran_serve(1, i) >= service1_Lowrange(1, 1) & ran_serve(1, i) <= service1_Upprange(1, 1))
            serve_time1(i) = service1_time(1,1);
        elseif(ran_serve(1, i) >= service1_Lowrange(1, 2) & ran_serve(1, i) <= service1_Upprange(1, 2))
            serve_time1(i) = service1_time(1,2);
        elseif(ran_serve(1, i) >= service1_Lowrange(1, 3) & ran_serve(1, i) <= service1_Upprange(1, 3))
            serve_time1(i) = service1_time(1,3);
        elseif(ran_serve(1, i) >= service1_Lowrange(1, 4) & ran_serve(1, i) <= service1_Upprange(1, 4))
            serve_time1(i) = service1_time(1,4);
        elseif(ran_serve(1, i) >= service1_Lowrange(1, 5) & ran_serve(1, i) <= service1_Upprange(1, 5))
            serve_time1(i) = service1_time(1,5);
        end
    end
```

The service time for counter 1 to 3 use the same concept as the inter-arrival time. The service time array for counter 1 to 3 is initialized with zeros and has the same size as the number of customers (cus_num). The difference is that instead of starting from the second customer (i = 2), the code iterates over each customer using the variable i, starting from the first customer (i = 1) and compares the range specified by service time lower and upper range.

Determine Arrival Time

```
%calc arrival time (previous arrival time add with current inter-arrival time)

arr_time(1) = 0;

for(i = 2:cus_num)
    arr_time(i) = [arr_time(i-1) + inter_arr(1,i)];
end
```

The variable arr_time is initialized as an array, and the first element is set to 0 to represent the arrival time of the first customer. The code then iterates over each customer using the variable i, starting from the second customer (i = 2). For each customer, the arrival time is calculated by adding the previous arrival time (arr_time(i-1)) with the current inter-arrival time (inter_arr(1, i)). The resulting arrival times are stored in the arr_time array. By the end of the loop, the arr_time array will contain the arrival times for each customer.

Implementation of Simulation Table

```
%-------------------------------------------------
%Initialise all to 0
%service 1 time begins
serve_begin1 = [zeros(1,cus_num)];
%service 1 time ends
serve_ends1 = [zeros(1,cus_num)];
%service 2 time begins
serve_begin2 = [zeros(1,cus_num)];
%service 2 time ends
serve_ends2 = [zeros(1,cus_num)];
%service 3 time begins
serve_begin3 = [zeros(1,cus_num)];
%service 3 time ends
serve_ends3 = [zeros(1,cus_num)];
%waiting time
wait_time = [zeros(1,cus_num)];
%time spent
time_spent = [zeros(1,cus_num)];
%serving time
serve_time_end = [zeros(1,cus_num)];
```

Initialization:
The code begins by initializing several arrays and variables to store information about customers, counters, and service times. These include arrays `serve_begin1`, `serve_ends1`, which means the service time begin and ends of counter 1, `serve_begin2`, `serve_ends2`(service time begin and ends of counter 2),`serve_begin3`, `serve_ends3`(service time begin and ends of counter 3) , `wait_time`(waiting time), `time_spent`(time spent in the system) , and `serve_time_end` (general service time end for ech customer). These arrays will be used to track various aspects of each customer's journey through the system.

```
%Table calculation
%-------------------------------------------------------------------------------------------------
%customer status(0 = NOT ARIIVE 1= ARRIVE AT QUEUE 2 = AT THE COUNTER, 3 = EXITED THE COUNTER)
cus_status = zeros(1,cus_num);
```

Customer Status:
The `cus_status` array is initialized with zeros to represent the status of each customer. The status can be one of the following:

- 0: Customer has not arrived.
- 1: Customer has arrived at the queue.
- 2: Customer is at the counter.
- 3: Customer has exited the counter.

```
%to determine which counter customer went
cus_counter_status = zeros(1,cus_num);

%counter status(0 = available, 1 = not available)
counter1_status = 0;
counter1_serviceEnd = zeros(1,cus_num);
counter2_status = 0;
counter2_serviceEnd = zeros(1,cus_num);
counter3_status = 0;
counter3_serviceEnd = zeros(1,cus_num);
```

Counter Status:
The `cus_counter_status` array is initialized with zeros to determine which counter each customer goes to. The status can be one of the following:
- 0: Not assigned to a counter.
- 1: Assigned to counter 1.
- 2: Assigned to counter 2.
- 3: Assigned to counter 3.

The `counter1_status`, `counter2_status`, and `counter3_status` variables are initialized to track the availability of each counter. The status can be one of the following:
- 0: Counter is available.
- 1: Counter is not available (currently serving a customer).

The `counter1_serviceEnd`, `counter2_serviceEnd`, and `counter3_serviceEnd` arrays are initialized to store the service end times for each customer at each counter.

```
%time initial set to 0
time = 0;



%control while loop
to_continue = 1;
```

Time and Loop Control:
- The `time` variable is initialized to 0 and will be used to track the simulation time.
- The `to_continue` variable is set to 1 to control the while loop that simulates the queuing system. It will be set to 0 when all customers have exited the system.

```
while (to_continue == 1)

 %set arriving at queue customer status to 1
     for(i= 1:cus_num)
         if(arr_time(i) == time)
             cus_status(i) = 1;
         end
     end
```

Simulation Loop:
The while loop runs as long as the `to_continue` variable is 1, indicating that there are still customers in the system.

Within the loop, there are several sections:
1. Set Arriving Customers:
- This section iterates through each customer and checks if their arrival time matches the current simulation time (`time`).
 - If a customer has arrived, their customer status(`cus_status`) is updated to 1, indicating that they have arrived at the queue.

```
%set customer status to 3 if service has ended and counter status to available
    for(i= 1:cus_num)
        if(counter3_serviceEnd(i) == time & cus_status(i) == 2) % if customer at counter and service ended
            counter3_status = 0;
            cus_status(i) = 3; %customer exit
            serve_ends3(1,i) = time;
            serve_time_end(1,i) = time;
            fprintf('Departure of customer%2.0f at minute%3.0f\n\n',[i time])
        end
        if(counter1_serviceEnd(i) == time & cus_status(i) == 2)
            counter1_status = 0;
            cus_status(i) = 3;
            serve_ends1(1,i) = time;
            serve_time_end(1,i) = time;
            fprintf('Departure of customer%2.0f at minute%3.0f\n\n',[i time])
        end
        if(counter2_serviceEnd(i) == time & cus_status(i) == 2)
            counter2_status = 0;
            cus_status(i) = 3;
            serve_ends2(1,i) = time;
            serve_time_end(1,i) = time;
            fprintf('Departure of customer%2.0f at minute%3.0f\n\n',[i time])
        end
    end
```

2. Customer Departures:
- This section checks if any customers at the counters (cus_status(i) ==2) have finished their service at the current simulation time (`time`).
 - If a customer has finished, their respective counter status is updated to 0 (available), their `cus_status` is updated to 3 (exited), and the relevant arrays (`serve_ends`, `serve_time_end`) are updated to record the departure time.
 - A message is printed to indicate the departure of the customer.

```
%set customer who arrive at queue to go to counter and change cus_status to 2
    for(i= 1:cus_num)
        if(cus_status(i) == 1) %if arrive at queue
            if(ran_item(1,i) <=3 & counter3_status ==0) %if item less than 3, go express counter if empty
                cus_status(i) = 2; %At counter
                cus_counter_status(i) = 3;
                counter3_status = 1; %counter 3 busy
                serve_begin3(1,i) = time;
                wait_time(i) = time - arr_time(i); %time here is same as time service begin
                counter3_serviceEnd(i) = time + serve_time3(1,i)+ ran_item(1,i); %servicetime plus number of items
                time_spent(i) = counter3_serviceEnd(i) - arr_time(i); % time spent in system = service end subtract arrival time

                fprintf('Arrival of customer%2.0f at minute%3.0f and queue at the counter%3.0f\n\n',[i arr_time(i) cus_counter_status(i)]);

                fprintf('Service for customer%2.0f start at minute%3.0f\n\n',[i time]);


        elseif(counter1_status == 0)
            cus_status(i) = 2;
            cus_counter_status(i) = 1;
            counter1_status = 1;
            serve_begin1(1,i) = time;
            wait_time(i) = time - arr_time(i);
            counter1_serviceEnd(i) = time + serve_time1(1,i) + ran_item(1,i);
            time_spent(i) = counter1_serviceEnd(i) - arr_time(i);

            fprintf('Arrival of customer%2.0f at minute%3.0f and queue at the counter%3.0f\n\n',[i arr_time(i) cus_counter_status(i)]);

            fprintf('Service for customer%2.0f start at minute%3.0f\n\n',[i time]);

        elseif(counter2_status == 0)
            cus_status(i) = 2;
            cus_counter_status(i) = 2;
            counter2_status = 1;
            serve_begin2(1,i) = time;
            wait_time(i) = time - arr_time(i);
            counter2_serviceEnd(i) = time + serve_time2(1,i)+ ran_item(1,i);
            time_spent(i) = counter2_serviceEnd(i) - arr_time(i);

            fprintf('Arrival of customer%2.0f at minute%3.0f and queue at the counter%3.0f\n\n',[i arr_time(i) cus_counter_status(i)]);

            fprintf('Service for customer%2.0f start at minute%3.0f\n\n',[i time]);
        end
    end
end
```

3. Customer Service:

- This section of code checks for customers (cus_status(i) == 1) who have arrived at the queue and assigns them to available counters based on certain conditions:

- If the customer has fewer than or equal to 3 items (ran_item(1,i) <= 3) and the express counter (counter3_status) is available (counter3_status == 0), the customer is assigned to the express counter (counter 3). The customer's status is set to 2 (at the counter), and the counter status and service end time are updated accordingly.
- If the express counter is not available, the customer is assigned to counter 1 if it is available (counter1_status == 0). The customer's status is updated, and the counter status and service end time are updated.
- If both the express counter and counter 1 are not available, the customer is assigned to counter 2 if it is available (counter2_status == 0). The customer's status is updated, and the counter status and service end time are updated.

-The customer status is updated to the respective counters' number, counter status will be updated to 1 (not available)

-The respective counters' service end time is calculated by adding the current simulation time which is also the service begin time with the service time for the respective counter and also the number of items acquired by the respective customer.

-It also calculates the waiting time (wait_time) by using the current simulation time which is also the service time begin deduct by the respective customer's arrival time.

–The time spent in the system (time_spent) is calculated by using the respective counter service end time deduct by the respective customer's arrival time.

- Messages are then printed to indicate the customer's arrival at the counter and the start of their service.

```
        time = time+1;
        counter = 0;

        for(i= 1:cus_num)
            if(cus_status(i) == 3 ) %count exited customer
                counter = counter + 1;
            end
        end

        if(counter == cus_num)
            to_continue = 0;
        end

    end
```

4. Time and Exit Check:
- The `time` variable is incremented by 1 to move to the next simulation time.
- The number of exited customers (`counter`) is counted to check if all customers have exited the system. If so, the `to_continue` variable is set to 0 to end the simulation loop.

The simulation continues until all customers have exited the system, and then the code will exit.

Evaluation of Results

```
%average wating time of a customer
total_wait = 0;
for(i=1:cus_num)
    total_wait = total_wait + wait_time(1,i);
end
avg_wait = total_wait / cus_num;
```

The code initializes a variable total_wait to 0 to store the sum of waiting times for all customers. It then iterates over each customer and adds their individual waiting time (wait_time(1, i)) to total_wait. Finally, it calculates the average waiting time by dividing total_wait by the total number of customers (cus_num).

```
%average inter-arrival time
total_inter_arr_time = 0;
for(i=2:cus_num)
    total_inter_arr_time = total_inter_arr_time + inter_arr(1,i);
end
avg_inter_arr_time = total_inter_arr_time / (cus_num-1);
```

The code initializes a variable total_inter_arr_time to 0 to store the sum of inter-arrival times for all customers except the first one. It then iterates over each customer starting from the second one and adds their individual inter-arrival time (inter_arr(1, i)) to total_inter_arr_time. Finally, it calculates the average inter-arrival time by dividing total_inter_arr_time by the number of inter-arrival intervals, which is one less than the total number of customers (cus_num-1).

```
%average time spent
total_time_spent = 0;
for(i=1:cus_num)
    total_time_spent = total_time_spent + time_spent(i);
end
avg_time_spent = total_time_spent / cus_num;
```

The code initializes a variable total_time_spent to 0 to store the sum of time spent by all customers. It then iterates over each customer and adds their individual time spent (time_spent(i)) to total_time_spent. Finally, it calculates the average time spent by dividing total_time_spent by the total number of customers (cus_num).

```
%probability that a customer has to wait in queue
num_cus_wait = 0;
for(i=1:cus_num)
    if(wait_time(i)> 0)
        num_cus_wait = num_cus_wait +1;
    end
end
prob_cus_wait = num_cus_wait / cus_num;
```

The code initializes a variable num_cus_wait to 0 to count the number of customers who had to wait. It then iterates over each customer and checks if their waiting time (wait_time(i)) is greater than 0. If the waiting time is greater than 0, it increments num_cus_wait. Finally, it calculates the probability by dividing num_cus_wait by the total number of customers (cus_num).

```
%average service time for each counter
total_serve1 = 0;
total_serve2 = 0;
total_serve3 = 0;
cus_num_counter1 =0;
cus_num_counter2 =0;
cus_num_counter3 =0;

for(i=1:cus_num)
    if(cus_counter_status(i) == 1)
        total_serve1 = total_serve1 + serve_time1(1, i) + ran_item(1,i);
        cus_num_counter1 = cus_num_counter1 + 1;
    elseif(cus_counter_status(i) ==2)
        total_serve2 = total_serve2 + serve_time2(1, i) + ran_item(1,i);
        cus_num_counter2 = cus_num_counter2 + 1;
    elseif(cus_counter_status(i) ==3)
        total_serve3 = total_serve3 + serve_time3(1, i) + ran_item(1,i);
        cus_num_counter3 = cus_num_counter3 + 1;
    end
end

avg_serve1 = total_serve1 / cus_num_counter1;
avg_serve2 = total_serve2 / cus_num_counter2;
avg_serve3 = total_serve3 / cus_num_counter3;
```

The code initializes variables total_serve1, total_serve2, and total_serve3 to 0 to store the sum of service times for each counter. It also initializes variables cus_num_counter1, cus_num_counter2, and cus_num_counter3 to 0 to count the number of customers served by each counter. It then iterates over each customer and checks their counter status (cus_counter_status(i)) to determine which counter they were served at. Depending on the counter status, it adds the corresponding service time (serve_time1, serve_time2, or serve_time3) plus the time taken for the item to be scanned (ran_item(1,i)) to the respective total_serve variable. It also increments the corresponding cus_num_counter variable to keep track of the number of customers served by each counter. Finally, it calculates the average service time for each counter by dividing the respective total service time by the corresponding number of customers served by the respective counter.

Output

RANDI Uniformly Distributed Integer

```
--> queue

Please choose type of random number generator
-----------------------------------------------
1 RANDI Uniformly Distributed Integer
2 RANDEXP Exponentially Distributed Integer
3 Linear Congruential Generators (LCG)
4 RAND Uniform Random Number Generator
->1

Please choose the number of customers for the simulation
---------------------------------------------------------
->6


COUNTER 1 SERVICE TIME PROBABILITY TABLE
-------------------------------------------------------------------------
Service time for Counter 1(min)| Probability | CDF  | Random number range
               3               |    0.21     | 0.21 |       1-21
               4               |    0.20     | 0.41 |      22-41
               5               |    0.21     | 0.62 |      42-62
               6               |    0.17     | 0.79 |      63-79
               7               |    0.21     | 1.00 |      80-100

COUNTER 2 SERVICE TIME PROBABILITY TABLE
-------------------------------------------------------------------------
Service time for Counter 2(min)| Probability | CDF  | Random number range
               3               |    0.16     | 0.16 |       1-16
               4               |    0.29     | 0.45 |      17-45
               5               |    0.18     | 0.63 |      46-63
               6               |    0.13     | 0.76 |      64-76
               7               |    0.24     | 1.00 |      77-100

COUNTER 3 SERVICE TIME PROBABILITY TABLE (EXPRESS COUNTER)
-------------------------------------------------------------------------
Service time for Counter 3(min)| Probability | CDF  | Random number range
               1               |    0.24     | 0.24 |       1-24
               2               |    0.26     | 0.50 |      25-50
               3               |    0.14     | 0.64 |      51-64
               4               |    0.16     | 0.80 |      65-80
               5               |    0.20     | 1.00 |      81-100
```

```
INTER ARRIVAL TIME PROBABILITY TABLE
-------------------------------------------------------------------------
Interarrival time(min)| Probability | CDF   | Random number range
           1          |    0.26     | 0.26  |       1-26
           2          |    0.20     | 0.46  |      27-46
           3          |    0.13     | 0.59  |      47-59
           4          |    0.17     | 0.76  |      60-76
           5          |    0.24     | 1.00  |      77-100

--------------------------------------
 RANDI Uniformly Distributed Integer
--------------------------------------

Exhibited Message:

Arrival of customer 1 at minute  0 and queue at the counter  1

Service for customer 1 start at minute  0

Arrival of customer 2 at minute  3 and queue at the counter  3

Service for customer 2 start at minute  3

Arrival of customer 3 at minute  5 and queue at the counter  2

Service for customer 3 start at minute  5

Departure of customer 2 at minute  7

Arrival of customer 4 at minute  7 and queue at the counter  3

Service for customer 4 start at minute  7

Departure of customer 4 at minute 12

Arrival of customer 5 at minute 11 and queue at the counter  3

Service for customer 5 start at minute 12

Departure of customer 3 at minute 14

Arrival of customer 6 at minute 12 and queue at the counter  2

Service for customer 6 start at minute 14
```

```
Departure of customer 1 at minute 15

Departure of customer 5 at minute 16

Departure of customer 6 at minute 25
```

N	RN Inter-arrival Time	Inter-arrival time	Arrival Time	Number of Items Acquired
1	-	-	0	10
2	49	3	3	2
3	28	2	5	4
4	36	2	7	1
5	66	4	11	3
6	26	1	12	8

Counter 1:

N	RN for Service Time	Service Time	Time Service Begins	Time Service Ends	Waiting Time	Time Spent in System
1	62	5	0	15	0	15

Counter 2:

N	RN for Service Time	Service Time	Time Service Begins	Time Service Ends	Waiting Time	Time Spent in System
3	60	5	5	14	0	9
6	2	3	14	25	2	13

Counter 3:

N	RN for Service Time	Service Time	Time Service Begins	Time Service Ends	Waiting Time	Time Spent in System
2	42	2	3	7	0	4
4	65	4	7	12	0	5
5	23	1	12	16	1	5

```
Evaluation Results of Simulation
---------------------------------
Average wating time of a customer :
    0.5000

Average inter-arrival time        :
    2.4000

Average time spent                :
    8.5000

Probability that a customer has to wait in queue:
    0.3333

Average service time for Counter1 :
 15

Average service time for Counter2 :
 10

Average service time for Counter3 :
    4.3333
```

RANDEXP Exponentially Distributed Integer

```
--> queue

Please choose type of random number generator
---------------------------------------------
1 RANDI Uniformly Distributed Integer
2 RANDEXP Exponentially Distributed Integer
3 Linear Congruential Generators (LCG)
4 RAND Uniform Random Number Generator
->2

Please choose the number of customers for the simulation
--------------------------------------------------------
->6


COUNTER 1 SERVICE TIME PROBABILITY TABLE
-------------------------------------------------------------------------
Service time for Counter 1(min)| Probability | CDF  | Random number range
               3               |    0.32     | 0.32 |       1-32
               4               |    0.15     | 0.47 |      33-47
               5               |    0.16     | 0.63 |      48-63
               6               |    0.15     | 0.78 |      64-78
               7               |    0.22     | 1.00 |      79-100

COUNTER 2 SERVICE TIME PROBABILITY TABLE
-------------------------------------------------------------------------
Service time for Counter 2(min)| Probability | CDF  | Random number range
               3               |    0.17     | 0.17 |       1-17
               4               |    0.27     | 0.44 |      18-44
               5               |    0.21     | 0.65 |      45-65
               6               |    0.15     | 0.80 |      66-80
               7               |    0.20     | 1.00 |      81-100

COUNTER 3 SERVICE TIME PROBABILITY TABLE (EXPRESS COUNTER)
----------------------------------------------------------------------------
Service time for Counter 3(min)| Probability | CDF  | Random number range
               1               |    0.24     | 0.24 |       1-24
               2               |    0.16     | 0.40 |      25-40
               3               |    0.22     | 0.62 |      41-62
               4               |    0.13     | 0.75 |      63-75
               5               |    0.25     | 1.00 |      76-100
```

```
INTER ARRIVAL TIME PROBABILITY TABLE
-------------------------------------------------------------------------
Interarrival time(min)| Probability | CDF  | Random number range
           1          |    0.16     | 0.16 |       1-16
           2          |    0.19     | 0.35 |      17-35
           3          |    0.16     | 0.51 |      36-51
           4          |    0.30     | 0.81 |      52-81
           5          |    0.19     | 1.00 |      82-100

-----------------------------------------
 RANDEXP Exponentially Distributed Integer
-----------------------------------------

Exhibited Message:

Arrival of customer 1 at minute  0 and queue at the counter  1

Service for customer 1 start at minute  0

Arrival of customer 2 at minute  2 and queue at the counter  3

Service for customer 2 start at minute  2

Departure of customer 2 at minute  6

Arrival of customer 3 at minute  7 and queue at the counter  3

Service for customer 3 start at minute  7

Arrival of customer 4 at minute  8 and queue at the counter  2

Service for customer 4 start at minute  8

Departure of customer 1 at minute 12

Arrival of customer 5 at minute 12 and queue at the counter  1

Service for customer 5 start at minute 12

Departure of customer 3 at minute 14

Departure of customer 4 at minute 20
```

```
Arrival of customer 6 at minute 15 and queue at the counter  2

Service for customer 6 start at minute 20

Departure of customer 5 at minute 23

Departure of customer 6 at minute 28
```

N	RN Inter-arrival Time	Inter-arrival time	Arrival Time	Number of Items Acquired
1	-	-	0	6
2	25	2	2	2
3	83	5	7	2
4	7	1	8	6
5	59	4	12	5
6	37	3	15	5

Counter 1:

N	RN for Service Time	Service Time	Time Service Begins	Time Service Ends	Waiting Time	Time Spent in System
1	72	6	0	12	0	12
5	65	6	12	23	0	11

Counter 2:

N	RN for Service Time	Service Time	Time Service Begins	Time Service Ends	Waiting Time	Time Spent in System
4	76	6	8	20	0	12
6	9	3	20	28	5	13

Counter 3:

N	RN for Service Time	Service Time	Time Service Begins	Time Service Ends	Waiting Time	Time Spent in System
2	34	2	2	6	0	4
3	97	5	7	14	0	7

```
Evaluation Results of Simulation
---------------------------------
Average wating time of a customer :
    0.8333

Average inter-arrival time        :
 3

Average time spent                :
    9.8333

Probability that a customer has to wait in queue:
    0.1667

Average service time for Counter1 :
   11.5000

Average service time for Counter2 :
 10

Average service time for Counter3 :
    5.5000
```

Linear Congruential Generator

```
--> queue

Please choose type of random number generator
---------------------------------------------
1 RANDI Uniformly Distributed Integer
2 RANDEXP Exponentially Distributed Integer
3 Linear Congruential Generators (LCG)
4 RAND Uniform Random Number Generator
->3

Please choose the number of customers for the simulation
--------------------------------------------------------
->6


COUNTER 1 SERVICE TIME PROBABILITY TABLE
------------------------------------------------------------------------
Service time for Counter 1(min)| Probability | CDF  | Random number range
               3               |    0.22     | 0.22 |       1-22
               4               |    0.21     | 0.43 |      23-43
               5               |    0.20     | 0.63 |      44-63
               6               |    0.17     | 0.80 |      64-80
               7               |    0.20     | 1.00 |      81-100

COUNTER 2 SERVICE TIME PROBABILITY TABLE
------------------------------------------------------------------------
Service time for Counter 2(min)| Probability | CDF  | Random number range
               3               |    0.22     | 0.22 |       1-22
               4               |    0.16     | 0.38 |      23-38
               5               |    0.25     | 0.63 |      39-63
               6               |    0.17     | 0.80 |      64-80
               7               |    0.20     | 1.00 |      81-100

COUNTER 3 SERVICE TIME PROBABILITY TABLE (EXPRESS COUNTER)
------------------------------------------------------------------------
Service time for Counter 3(min)| Probability | CDF  | Random number range
               1               |    0.32     | 0.32 |       1-32
               2               |    0.17     | 0.49 |      33-49
               3               |    0.16     | 0.65 |      50-65
               4               |    0.16     | 0.81 |      66-81
               5               |    0.19     | 1.00 |      82-100
```

INTER ARRIVAL TIME PROBABILITY TABLE

Interarrival time(min)	Probability	CDF	Random number range
1	0.19	0.19	1-19
2	0.15	0.34	20-34
3	0.16	0.50	35-50
4	0.18	0.68	51-68
5	0.32	1.00	69-100

```
-------------------------------------
    Linear Congruential Generator
-------------------------------------

Exhibited Message:

Arrival of customer 1 at minute  0 and queue at the counter  3

Service for customer 1 start at minute  0

Arrival of customer 2 at minute  4 and queue at the counter  1

Service for customer 2 start at minute  4

Departure of customer 1 at minute  7

Arrival of customer 3 at minute  7 and queue at the counter  2

Service for customer 3 start at minute  7

Departure of customer 2 at minute 13

Arrival of customer 4 at minute 12 and queue at the counter  1

Service for customer 4 start at minute 13

Arrival of customer 5 at minute 13 and queue at the counter  3

Service for customer 5 start at minute 13

Departure of customer 3 at minute 14

Departure of customer 5 at minute 16
```

```
Arrival of customer 6 at minute 18 and queue at the counter  2

Service for customer 6 start at minute 18

Departure of customer 4 at minute 25

Departure of customer 6 at minute 31
```

N	RN Inter-arrival Time	Inter-arrival time	Arrival Time	Number of Items Acquired
1	-	-	0	2
2	54	4	4	4
3	39	3	7	4
4	93	5	12	5
5	14	1	13	1
6	79	5	18	7

Counter 1:

N	RN for Service Time	Service Time	Time Service Begins	Time Service Ends	Waiting Time	Time Spent in System
2	50	5	4	13	0	9
4	83	7	13	25	1	13

Counter 2:

N	RN for Service Time	Service Time	Time Service Begins	Time Service Ends	Waiting Time	Time Spent in System
3	3	3	7	14	0	7
6	80	6	18	31	0	13

Counter 3:

N	RN for Service Time	Service Time	Time Service Begins	Time Service Ends	Waiting Time	Time Spent in System
1	83	5	0	7	0	7
5	40	2	13	16	0	3

```
Evaluation Results of Simulation
--------------------------------
Average wating time of a customer :
    0.1667

Average inter-arrival time        :
    3.6000

Average time spent                :
    8.6667

Probability that a customer has to wait in queue:
    0.1667

Average service time for Counter1 :
   10.5000

Average service time for Counter2 :
 10

Average service time for Counter3 :
 5
```

RAND Uniform Random Number Generator

```
--> queue

Please choose type of random number generator
---------------------------------------------
1 RANDI Uniformly Distributed Integer
2 RANDEXP Exponentially Distributed Integer
3 Linear Congruential Generators (LCG)
4 RAND Uniform Random Number Generator
->4

Please choose the number of customers for the simulation
--------------------------------------------------------
->6


COUNTER 1 SERVICE TIME PROBABILITY TABLE
------------------------------------------------------------------------
Service time for Counter 1(min)| Probability | CDF  | Random number range
               3               |    0.27     | 0.27 |       1-27
               4               |    0.15     | 0.42 |      28-42
               5               |    0.17     | 0.59 |      43-59
               6               |    0.19     | 0.78 |      60-78
               7               |    0.22     | 1.00 |      79-100

COUNTER 2 SERVICE TIME PROBABILITY TABLE
------------------------------------------------------------------------
Service time for Counter 2(min)| Probability | CDF  | Random number range
               3               |    0.19     | 0.19 |       1-19
               4               |    0.29     | 0.48 |      20-48
               5               |    0.16     | 0.64 |      49-64
               6               |    0.17     | 0.81 |      65-81
               7               |    0.19     | 1.00 |      82-100

COUNTER 3 SERVICE TIME PROBABILITY TABLE (EXPRESS COUNTER)
------------------------------------------------------------------------
Service time for Counter 3(min)| Probability | CDF  | Random number range
               1               |    0.27     | 0.27 |       1-27
               2               |    0.16     | 0.43 |      28-43
               3               |    0.15     | 0.58 |      44-58
               4               |    0.19     | 0.77 |      59-77
               5               |    0.23     | 1.00 |      78-100
```

```
INTER ARRIVAL TIME PROBABILITY TABLE
-------------------------------------------------------------------------------
Interarrival time(min)| Probability | CDF  | Random number range
           1          |    0.13     | 0.13 |       1-13
           2          |    0.16     | 0.29 |      14-29
           3          |    0.22     | 0.51 |      30-51
           4          |    0.22     | 0.73 |      52-73
           5          |    0.27     | 1.00 |      74-100

--------------------------------------
 RAND Uniform Random Number Generator
--------------------------------------

Exhibited Message:

Arrival of customer 1 at minute  0 and queue at the counter  1

Service for customer 1 start at minute  0

Arrival of customer 2 at minute  4 and queue at the counter  2

Service for customer 2 start at minute  4

Arrival of customer 4 at minute 12 and queue at the counter  3

Service for customer 4 start at minute 12

Departure of customer 1 at minute 15

Arrival of customer 3 at minute  9 and queue at the counter  1

Service for customer 3 start at minute 15

Departure of customer 2 at minute 17

Departure of customer 4 at minute 17

Arrival of customer 5 at minute 16 and queue at the counter  3

Service for customer 5 start at minute 17

Arrival of customer 6 at minute 20 and queue at the counter  2
```

```
Service for customer 6 start at minute 20

Departure of customer 5 at minute 25

Departure of customer 3 at minute 29

Departure of customer 6 at minute 32
```

N	RN Inter-arrival Time	Inter-arrival time	Arrival Time	Number of Items Acquired
1	-	-	0	9
2	68	4	4	8
3	99	5	9	9
4	30	3	12	2
5	54	4	16	3
6	55	4	20	8

Counter 1:

N	RN for Service Time	Service Time	Time Service Begins	Time Service Ends	Waiting Time	Time Spent in System
1	71	6	0	15	0	15
3	52	5	15	29	6	20

Counter 2:

N	RN for Service Time	Service Time	Time Service Begins	Time Service Ends	Waiting Time	Time Spent in System
2	58	5	4	17	0	13
6	42	4	20	32	0	12

Counter 3:

N	RN for Service Time	Service Time	Time Service Begins	Time Service Ends	Waiting Time	Time Spent in System
4	55	3	12	17	0	5
5	94	5	17	25	1	9

```
Evaluation Results of Simulation
---------------------------------
Average wating time of a customer :
    1.1667

Average inter-arrival time        :
 4

Average time spent                :
   12.3333

Probability that a customer has to wait in queue:
    0.3333

Average service time for Counter1 :
   14.5000

Average service time for Counter2 :
   12.5000

Average service time for Counter3 :
    6.5000
```