MIDDLE EAST TECHNICAL UNIVERSITY, NORTHERN CYPRUS CAMPUS
CNG213 Data Structures – Programming Assignment 2

Date handed out: **02 May 2021, Sunday**
Date submission due: **16 May 2021, Sunday, 23:55 (Cyprus Time)**

Delivery Simulator

Today's consumers expect fast delivery wherever they shop online. Hence, sellers across marketplaces need to keep up with customer expectations by providing fast delivery to offer a great customer experience.

This assignment aims to help you to practice queue and priority queue abstract data types. You will write a simulator that will help sellers across marketplaces to simulate a queue to gather some statistics.

For a seller, there is a single queue and all customers are located in this queue when they purchase an item and request a delivery. There are three types of delivery and each have a different level of priority based on their type. Table 1 shows the delivery types with their priority values where the highest priority is 3 and the lowest priority is 1.

Table 1. Delivery Types

Patient Type	Priority
Express (E)	3
Standard (S)	2
Free (F)	1

You will need to create and maintain a priority queue for the customers. They will be served based on the chosen delivery priority and also their purchase time. Let the following queue is the current queue of the seller.

Front | E E S S F | Rear

When a new customer purchases an item and choose express delivery type, s/he is located in the third position of the queue (after the first two express delivery) because his/her priority is higher than the customers who have chosen standard or free delivery. Therefore, the queue will be as follows:

Front | E E **E** S S F | Rear

First of all, you need to create a list of customers that will purchase an item and request delivery. For each customer, you need to keep the track of the following:

- Delivery type (E, S or F) – this will represent their priority;
- Purchase time (in minutes, such as at 15th minute) – this will represent when customer purchase an item and request for delivery;
- Delivery time (in minutes) – this will represent how long will it take to deliver the purchased item to the customer;
- Delivery start time (in minutes) – this will represent the actual time that the courier will start to deliver this particular purchased item to the customer;
- Courier ID – this will represent the ID of the courier that will do the delivery (the ID of the first courier is 1, the ID of the second courier is 2, … so) – This is to be able to represent the possibility that the seller might have more than one courier.
- Amount of purchase– this will represent the amount of the payment. If the amount of purchase is greater than or equal to 500 £, it will be category A. Otherwise, it will be category B.

The number of customers, the number of couriers, the maximum purchase time, and the maximum delivery time should be provided as an input at the beginning of the program.

You need to name your program as **`deliverySimulator`** and needs to run on a command line by using the following format:

```
deliverySimulator noOfCustomers noOfCouriers maxPurchaseTime maxDeliveryTime
```

An example of a customer list created with the following input is shown in Table 2.

```
deliverySimulator 5 2 20 5
```

Based on the given input, there are 5 customers who purchased an item and request for delivery, and 2 couriers. The maximum purchase time is 20 and the maximum delivery time is 5. Once you get this input data from the user, then you need to prepare your simulation data based on this given input. At the beginning, the fields of delivery start time and courier ID are equal to 0, because you need to update these fields after the item is delivered to the customer by a courier. Other fields will be prepared as follows: Since in the given example, you have 5 customers, you need to prepare simulation data for 5 customers based on the parameters given above. Please see example data prepared in Table 2. You need to use a randomiser for purchase time and delivery time generation. However, you need to remember that the purchase time needs to be less than maximum purchase time given by the user and the delivery time needs to be less than the maximum delivery time.

You also need to randomly generate a type for customer's requested delivery types which can be Express (E), Standard (S) and Free (F). You also need to randomly assign the amount of purchase. You can see an example data in Table 2 which is prepared based on the input above.

Table 2. An example of a customer list

Customer Type	**Purchase Time**	**Delivery Time**	**Delivery Start Time**	**Courier**	**Amount of purchase**
E	5	3	0	0	A
F	7	5	0	0	B
F	8	5	0	0	B
S	15	2	0	0	B
E	20	3	0	0	A

When your customer list is ready, you need to create an empty queue and also an integer array to keep the availability of the couriers (0: Busy, 1: Available). For example, if there are two couriers, you need to have an array of 2 integers. As these couriers are available at the beginning, you need to initialise the array with 1s.

Once these are ready then you need to start processing the customers which are prepared in the customers' list. When a customer needs to be served at his/her purchase time (note that you generated them randomly), you need to check the current state of the queue.

- If nobody is waiting in the queue, you need to randomly assign the customer to one of the couriers to be served and you need to make the availability of that courier "Busy" and then update the delivery start time for the customer.
- If the queue is not empty, you need to locate the customer in the appropriate position in the queue (see an example above).

When an item is delivered to a customer by a particular courier, the courier ID should be updated for the customer. After that, the availability of the courier should be converted to "Available". When a courier is available for next customer and there is another customer in the queue, the customer should be assigned to that courier. When more than one courier is available at the same time, you need to randomly assign the first customer in the queue to one of the available couriers. When all the customers are served, the simulation will be completed.

You will also need to maintain a simulated clock which keeps the time of the latest event handled. There are three types of events which are as follows: (1) customer purchase, (2) start delivery and (3) complete delivery. At the beginning, the simulated clock should be equal to 0. Regarding the simulated clock, two examples of scenarios are given below.

- If the first event is "customer purchase" at the 5th minute, then you need to advance the simulated clock to 5.
- If the current clock is 21 and the nearest event is "complete delivery" at the 25th minute, then you need to advance the simulated clock to 25.

When the simulation is completed, you need to report the following information/statistics:

- The number of couriers: How many couriers served the customers? (This is already given to you by the user)
- The number of customers: How many customers were served? (This is already given to you by the user)
- The number of customers for each delivery type: How many customers are served by the couriers with Express, Standard and Free delivery type?
- The number of customers for each courier: How many customers are served by each courier?
- The completion time: How long did it take to complete the simulation?
- Average time spent in the queue: What was the average delay in the queue? You need to calculate the total waiting time of all customers in the queue and then divide this by the number of customers you have. This will give you the average waiting time in the queue.
- Maximum waiting time: What was the longest wait time in the queue? You need to find the customer who waited longest in the queue.
- Popular Purchase: This will display the purchase type mostly done by the customers.

When a customer is served by a courier, you need to put it back to the list of customers that is created at the beginning of the program. Then you need to use the list to provide the required information/statistics.

Programming Requirements:

In this assignment, you are expected to write the following functions:

- **`parseInput:`** This function should parse the input and set the values of the number of customers, the number of couriers, the maximum purchase time and the maximum delivery time.
- **`createCustomerList:`** This function should randomly create customers based on the input (the number of customers, the number of couriers, the maximum purchase time, the maximum delivery time). The customers should be stored in a linked list in ascending order based on their purchase time.
- **`initialiseSimulator:`** This function should create an empty queue, and also an integer array to keep the availability of the couriers.
- **`newCustomer:`** This function takes a customer from the customer list based on the purchase time and add him/her to the queue.
- **`serveCustomer:`** This function takes a customer from the queue to be served by a courier.
- **`reportStatistics:`** This function reports the statistics, such as the average time spent in the queue, maximum waiting time, etc. (see above).
- **`main:`** The main function is responsible to coordinate all the functions, simulated clock and other required operations to run the simulator successfully.

Submission Requirements:

In this assignment, you need to have a header file (queue.h) which includes the major functionality of the Queue ADT. If you will use other ADTs, you need to create a separate header file for each of them. You also need to have a C source file (deliverySimulator.c) that includes the main function and other functions. You need put all these files into the "cng213a2" folder and then submit the compressed version of the folder to ODTU-CLASS.

Grading Policy:

Grading Item	**Mark (out of 100)**
Data Structures	5
Input	5
Function: **`parseInput`**	5
Function: **`createCustomerList`**	10
Function: **`initialiseSimulator`**	5
Function: **`newCustomer`**	20
Function: **`serveCustomer`**	10
Function: **`reportStatistics`**	10
Function: **`main`**	25
Submission requirements followed	5

Important Notes:

- Remember to have good programming style (Appropriate comments, variable names, formulation of selection statements and loops, reusability, extensibility etc.). Each of the items above will include10% for good programming style.
- Read rules regarding to assignments from the Syllabus carefully.
- If your code does not compile due to syntax errors, you will automatically get zero.
- If your code includes a variable declaration inside a for loop such as for(int i=0; i<5;i++), you will automatically get zero.
- If your code includes global varibles, you will automatically get zero.

Sample run could be as follows:

```
****************Delivery Statistics*****************
The number of couriers: 2
The number of customers: 5
Number of customers for each delivery type:
      Express: 2
      Standard: 1
      Free: 2
Number of customers for each courier:
      Courier 1:4
      Courier 2:1
```

```
Completion time: 23
Average time spent in the queue: 1.8
Maximum waiting time: 5
Popular purchase: B
```