Chandigarh College of Engineering & Technology (Degree Wing)
Department of Computer Science & Engineering

REPORT Problem-6

Abdul Rahim (CO20301)

Files included:

queue.c, report

Data structure used:

queue implementation of doubly linked list

Outputs: Following are the screenshots representing the outputs of different sets of data used in each method. The program was executed on VScode

Insert_in_queue() method:

```
This hypothetical Queue describes a queue of people in front of a movie threater.
The money each person has is described by the data.


Enter 1 to insert in queue.

Enter 2 to delete from queue.

Enter 3 to search in queue.

Enter 4 to update into queue.

Enter 5 to display queue.

Enter 6 to check if queue is full.

Enter 7 to check if queue is empty.

Enter 8 to EXIT PROGRAM.

Enter your choice: 1

You entered: 1

Enter the money this person has: 300
Person inserted in queue. Total no. of people in quque now: 1
```

```
This hypothetical Queue describes a queue of people in front of a movie threater.
The money each person has is described by the data.


Enter 1 to insert in queue.

Enter 2 to delete from queue.

Enter 3 to search in queue.

Enter 4 to update into queue.

Enter 5 to display queue.

Enter 6 to check if queue is full.

Enter 7 to check if queue is empty.

Enter 8 to EXIT PROGRAM.

Enter your choice: 1

You entered: 1

Enter the money this person has: 270
Person inserted in queue. Total no. of people in quque now: 2
```

```
This hypothetical Queue describes a queue of people in front of a movie threater.
The money each person has is described by the data.


Enter 1 to insert in queue.

Enter 2 to delete from queue.

Enter 3 to search in queue.

Enter 4 to update into queue.

Enter 5 to display queue.

Enter 6 to check if queue is full.

Enter 7 to check if queue is empty.

Enter 8 to EXIT PROGRAM.

Enter your choice: 1

You entered: 1

Enter the money this person has: 500
Person inserted in queue. Total no. of people in quque now: 3
```

Display:

```
This hypothetical Queue describes a queue of people in front of a movie threater.
The money each person has is described by the data.


Enter 1 to insert in queue.

Enter 2 to delete from queue.

Enter 3 to search in queue.

Enter 4 to update into queue.

Enter 5 to display queue.

Enter 6 to check if queue is full.

Enter 7 to check if queue is empty.

Enter 8 to EXIT PROGRAM.

Enter your choice: 5

You entered: 5

Queue position: 1, Money :300
Queue position: 1, Money :270
Queue position: 1, Money :500
```

Zoomed in view:

```
Queue position: 1, Money :300
Queue position: 2, Money :270
Queue position: 3, Money :500
```

Delete_from_queue() method:

```
This hypothetical Queue describes a queue of people in front of a movie threater.
The money each person has is described by the data.


Enter 1 to insert in queue.

Enter 2 to delete from queue.

Enter 3 to search in queue.

Enter 4 to update into queue.

Enter 5 to display queue.

Enter 6 to check if queue is full.

Enter 7 to check if queue is empty.

Enter 8 to EXIT PROGRAM.

Enter your choice: 2

You entered: 2

First person was removed from the queue.
No. of people in queue now: 2.
```

```
Queue position: 1, Money :270
Queue position: 2, Money :500
```

Update:

```
This hypothetical Queue describes a queue of people in front of a movie threater.
The money each person has is described by the data.


Enter 1 to insert in queue.

Enter 2 to delete from queue.

Enter 3 to search in queue.

Enter 4 to update into queue.

Enter 5 to display queue.

Enter 6 to check if queue is full.

Enter 7 to check if queue is empty.

Enter 8 to EXIT PROGRAM.

Enter your choice: 4

You entered: 4

Note: You cannot access positions below 1 or more than end Of Queue which is 2
Enter the data(money this person has): 1000
Enter the position you want to update: 1
Updated successfully.
```

```
Queue position: 1, Money :1000
Queue position: 2, Money :500
```

Is full method():

```
This hypothetical Queue describes a queue of people in front of a movie threater.
The money each person has is described by the data.


Enter 1 to insert in queue.

Enter 2 to delete from queue.

Enter 3 to search in queue.

Enter 4 to update into queue.

Enter 5 to display queue.

Enter 6 to check if queue is full.

Enter 7 to check if queue is empty.

Enter 8 to EXIT PROGRAM.

Enter your choice: 6

You entered: 6

Queue is not full.
```

Is empty method();

After two delete queue was empty:

```
PS C:\Users\Abdul\Desktop\programs> cd "c:\Users\Abdul\Desktop\programs\assignment 6\" ;

This hypothetical Queue describes a queue of people in front of a movie threater.
The money each person has is described by the data.


Enter 1 to insert in queue.

Enter 2 to delete from queue.

Enter 3 to search in queue.

Enter 4 to update into queue.

Enter 5 to display queue.

Enter 6 to check if queue is full.

Enter 7 to check if queue is empty.

Enter 8 to EXIT PROGRAM.

Enter your choice: 7

You entered: 7

Queue is empty.
```

Conclusion:

Through the implementation of this program, we learned to perform various operation on the Queue data structure using doubly linked list. We learned how to link each method to each other and then finally to the main / control block of the program execution by working and designing Block diagrams and algorithms for each and every method. We learned how to implement Modular programming into our daily practice programs. We designed algorithms for every method before their implementation which made us think naturally rather than syntactically. For betterment, one could take a deep dive into the edge cases related to this or similar programs and debug those cases with a will to learn more and more.