AKFA UNIVERSITY

Algorithms and Data Structures
BS-III Fall 2022
LAB NO: 06

Name	
ID	
Date	

Objectives:

After completing this Lab students will able to

1. Implement Queue using Array.
2. Implement Queue using linked lists.
3. Implement priority Queues like structure (Priority queue is implemented using heapsort in actual)

For details and explanation of Queue ADT, please refer to the recommended book or lecture provided. You are supposed to perform the following tasks

1. Implement circular queues using Array. For array implementation, you need the following class

```
class QueueType
{
public:
bool isEmptyQueue();
bool isFullQueue();
void initializeQueue();
Type front();
Type back();
void addQueue(const Type& queueElement);
void deleteQueue();
queueType(int queueSize = 100); // Function with default parameters
queueType(const QueueType& otherQueue);
// To create a Queue object that copies another queue
~queueType();
private:
int maxQueueSize; int count;
int queueFront;
int queueRear;
Type *list;
};
```

Implement all the above methods and verify your results using the following code

```
QueueType Q1;
Q1.addQueue(15);
Q1.addQueue(25);
Q1.addQueue(4);
Q1.addQueue(12);
Q1.addQueue(123);
Q1.addQueue(75);
Q1.addQueue(85);
Q1.addQueue(55);
Q1.front(); // should display 15
Q1.deleteQueue();
```

```
Q1.printQueue();// should display all the queue element in the order they are added
except 15
	queueType Q2(Q1);
	Q1.addQueue(1000);
	Q1.addQueue(2000);
	Q1.printQueue(); // Should display all the members of Q1 in the order they are
inserted except 15 as it was deleted earlier
	Q2.printQueue();
	// Should display all the members of Q2 i-e all the members of Q1 except for 1000
and 2000
	Q1.initializeQueue();
	Q1.printQueue();
	//Should display a message "An empty Queue can't be displayed"
	Q1.addQueue(35);
	cout<<Q1.front()<<endl;
	//Should display 35
	cout<<Q1.front()<<endl;
	//Should display 35
	Q1.addQueue(3);
	Q1.printQueue();
	//Should display both 35 and 3
```

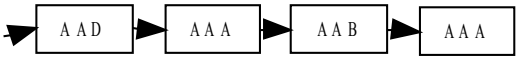
2. Implement queue using linked lists. Most of you have already implemented in previous labs, so this is a bonus hit for them. Provide the same instructions as in Task 1 to validate your queue.
3. Although, priority queues are implemented using Heapsort, however, lets try to implement such similar structure without thinking the complexity of the algorithm for a moment. Implement a priority queue using linked list for hospital administration where patients are categorized into the following priorities

   a. Normal patients with Priority 0
   b. Patient that pay double fee with priority 1
   c. Patients who can't wait longer due to some diseases such as neurological issues with priority 2
   d. Serious patients with priority 3

   You should implement all the functions, however, addQueue function should follow the priority. For example, if we have the following instructions

```
queueType<string> Q1;
Q1.addQueue("AAA", 0);
Q1.addQueue("AAB", 0);
Q1.addQueue("AAC", 0);
Q1.addQueue("AAD", 3);
```

   The queue should look like



   If a few more instructions are provided, such as

```
Q1.addQueue("AAE", 0);
Q1.addQueue("AAF", 3);
Q1.addQueue("AAG", 2);
```

AKFA UNIVERSITY

Algorithms and Data Structures
BS-III Fall 2022
LAB NO: 06

deleteQueue function should simply delete the first node. You should use both front and rare pointers.

Note: For this task, you have to use ordered linked list logic. For understanding of ordered linked list, please reach the following article
https://runestone.academy/ns/books/published/cppds/LinearLinked/ImplementinganOrderedList.html

In case if you have trouble understanding the concepts of the ordered linked list, do call me anytime.