Lesson – 9 - Lab Homework – Stack and Queue

This problem was asked in the previous batch Common Programming Test.

<u>Problem 1</u>. [Data Structures - Queue] For this problem, you will implement a queue of `ints`, using an array in the background. To start, initialize an array and two pointers (`front` and `rear`) in your queue class as follows:

```
private int[] arr = new int[10];
private int front = -1;
private int rear = 0;
```

When adding an element to the queue, you add the element to the position pointed to by the variable `rear`. When removing an element from the queue, you remove the element in the position pointed to by the variable `front`.

Implement all of the methods declared below so that your class behaves as a queue. The methods to be implemented are: `isEmpty, size, enqueue, dequeue`, and `peek`. A start-up class `ArrayQueueImpl`, listing all of these methods, has been provided for you. You will need to implement the methods in that class. Write a `Main` class with an implemented `main` method to test your own code.

Your code must meet the following requirements:

1. The `enqueue` method should never cause an exception to be thrown. In particular, your queue must support unlimited `enqueue(insertion/add)` operations. This means that you will need to incorporate a procedure for resizing the background array periodically.
2. The only situation in which `dequeue(delete/remove)` or `peek` should cause an exception to be thrown is if either of these operations is called when the queue is empty. In that case return -1also display the message in console as "Queue is Empty"
3. You may not use Java library collection classes to create your queue.
4. There must not be any compilation errors or runtime errors in the solution that you submit.

```
public class ArrayQueueImpl {

        private int[] arr = new int[10];
        private int front = -1;
        private int rear = 0;


        public int peek() {
                return -1;
        //implement
    }

    public void enqueue(int obj){
        //implement
    }

    public int dequeue() {
        return -1;
        //implement
    }

    public boolean isEmpty(){
        //implement
        return false;
    }

    public int size(){
        //implement
        return 0;
    }
    private void resize(){
        //implement
    }
}
```

Problem 2. [Data Structures - Stack]

Take the demo code ArrayStackDemo.java. In that file Stack is implemented using Array. Do the same behaviors by implementing stack using Linked List.[Either using singly linked list / Doubly linked list]