CS 2401 Assignment #8

Due Date: Sunday, April 16, 2017 11:59PM (See the syllabus for late policy).

Objective: The goal of this assignment is to practice Queue operations.

Assignment: During the lecture sessions, we discussed an array-based `Queue` class. The Queue class from the lecture is copied below.

```
public class Queue{
        private int QUEUE_SIZE = 50;
        private Object[] items;
        private int front, back, count;

        public Queue() {
                items = new Object[QUEUE_SIZE];
                front = 0;
                back = QUEUE_SIZE -1;
                count =0;
        }

        public boolean isEmpty(){
                return count ==0;
        }

        public boolean isFull(){
                return count == QUEUE_SIZE;
        }

        public void enqueue(Object newItem){
                if (!isFull()){
                        back = (back+1) % QUEUE_SIZE;
                        items[back] = newItem;
                        count++;
                  return;
                } else
                        System.out.println("Trying to enqueue into full queue");
        }

        public Object dequeue(){
                if (!isEmpty()){
                        Object queueFront = items[front];
                        front = (front+1) % QUEUE_SIZE;
                        count--;
                  return queueFront;
                }else
                        System.out.println("Trying to dequeue from empty queue");
                return null;
        }

        public void dequeueAll(){
                items = new Object[QUEUE_SIZE];
                front = 0;
                back = QUEUE_SIZE -1;
                count =0;
        }

        public Object peek(){
                if (!isEmpty()) {
```

```
                    return items[front];
            }
            else
                    System.out.println("Trying to peek with empty queue");
            return null;
        }

        public int size(){
            return count;
        }

}// End of class Queue
```

Your tasks in this assignment are outlined below.

1. Change the `QUEUE_SIZE=50` to `QUEUE_SIZE=5` in the given code. This would make the demo easier.

2. Change the `enqueue` method of the `Queue` class in such a way that if the array is full then the array-size will become double. Obviously, the new item will be added in the expanded array in that case. That is, `enqueue` will never fail due to the size-limitation of the array.

3. Write a different class named `Runner.java` from which you will create a queue object and demonstrate that your `Queue` class works. In `Runner.java`, in addition to the `main` method, write the following methods and demonstrate that these methods work.

    (a) **`public static void printQueue(Queue q):`** Print all the elements of a queue. We discussed this in the class.

    (b) **`public static void reverseQueue(Queue q):`** Reverse the content of the queue.

A sample terminal output of `Runner.java` is provided below.

```
My queue is as follows:
10 20 30 40 50
I am going to dequeue one element.
My queue is as follows:
20 30 40 50
I am going to reverse my Queue.
My queue is as follows:
50 40 30 20
I am going to enqueue 60.
My queue is as follows:
50 40 30 20 60
I am going to enqueue 70.
Queue is full. Doubling the size.
New max. size is: 10
Entered the new item.
My queue is as follows:
50 40 30 20 60 70
I am going to reverse my Queue.
My queue is as follows:
70 60 20 30 40 50
```

Deliverables: `Queue.java` and `Runner.java`. You must use Blackboard to submit. Talk to your TA for further instructions.