CS 2114 Practice Problems 3/14/2016

Part 1: Queues

1. A queue is _____ in _____ out.

2. Given the following implementations of enqueue and dequeue, what would the console display for each sequence of calls?

```
public void enqueue(T newEntry) {
      System.out.print(newEntry);
      Node newNode = new Node(newEntry, null);
      if (isEmpty()) {
            firstNode = newNode;
      } else {
            lastNode.setNextNode(newNode);
      }
      lastNode = newNode;
}

public T dequeue() {
      T front = getFront();
      if (firstNode == null) {
            System.out.print(" ERROR! ");
            return null;
      }
      firstNode.setData(null);
      firstNode = firstNode.getNextNode();
      if (firstNode == null){
            lastNode = null;
      }
      System.out.print(front);
      return front;
}
```

a. enqueue("A"); enqueue("B"); enqueue("C");

b. enqueue("A"); enqueue("B"); enqueue("C"); dequeue(); dequeue(); dequeue(); dequeue();

c. dequeue(); enqueue("A"); enqueue("B"); enqueue("C"); dequeue();

d. enqueue("A"); enqueue("B"); enqueue(dequeue()); enqueue(dequeue()); dequeue(); dequeue(); dequeue(); dequeue();

3. Given the class variables firstNode and lastNode, implement the method *isEmpty* that takes no parameters and returns a boolean.

4. Given the class variables firstNode and lastNode, implement the method *clear* that takes no parameters and does not return anything.

Part 2: Deques

1. Implement a *makeCircular* method that takes no parameters and does not return anything. A circular deque is when you can access the bottom of the queue via the top and visa versa. **Refer to the lecture slides to see what methods and fields are available.**

2. Below is the *removeBack* method for deques.

```
public T removeBack() {
      T back = getBack(); // May throw EmptyQueueException
      assert lastNode != null;
      lastNode = lastNode.getPreviousNode();
      if (lastNode == null)
            firstNode = null;
      else
            lastNode.setNextNode(null);
      return back;
}
```

Notice there is a call to the method *getBack*. Implement the method *getBack* that takes no parameters, returns an object of type T, and throws an EmptyQueueException when the queue is empty.