CSE 274, Fall 2018
Project 05, 60 points, Due: Sunday, October 21 by 11:59 pm

Outcomes:

- Implement an array-based ADT (queue).
- Implement a circular-array-based strategy.

Description:

The array-based implementations of the ADT queue in chapter 11 used a circular array. One implementation counted the entries in the queue, while the other left one location in the array unused. We used these strategies to tell when the queue was empty and when it was full.

A third strategy is possible. It does not count and does not have an unused location in the circular array. After initializing *frontIndex* to *0* and *backIndex* to *-1,* you do not use modulo arithmetic when you increment these fields. Instead, you use modulo arithmetic when you index the array, but without changing *frontIndex* and *backIndex*. Thus, if queue is the array, *queue[frontIndex % queue.length]* is the front entry, and the entry at the back of the queue is *queue[backIndex % queue.length]*.

Now if *backIndex* is less than *frontIndex*, the queue is empty. The number of entries in the queue is *backIndex – frontIndex + 1*. You can compare this number with the size of the array to see whether the array is full.

Since *frontIndex* and *backIndex* can continue to grow, they might become too large to represent. To reduce the chance of this happening, set *frontIndex* to *0* and *backIndex* to *-1* whenever the implementation detects an empty queue. Note that adding to a full queue invokes ensureCapacity, which sets *frontIndex* to *0* and *backIndex* to the index of the entry at the back of the queue.

Complete this array-based implementation of the ADT queue.

Naming requirements:

- You have been provided three source code files that you must download and put in your project:
    - QueueIntereface.java, which should not be changed in ANY WAY.
    - EmptyQueueException.java, which should not be changed in ANY WAY.
    - ArrayQueue.java, you need to complete all the methods in this class following the strategy described above.
    - Driver.java, which tests your ArrayQueue class. You can use or change this file to test the ArrayQueue class.

Grading Rubric:

You need to complete 9 methods in the ArrayQueue.java class. Each method has equal weight with a total of 60 points.