

Programming Assignment 9-3

For this problem, you will implement a queue of `ints`, using an array in the background. To start, use the following startup code:

```
public class ArrayQueueImpl {
    private int[] arr = new int[10];
    private int size = 0;
    private int front = -1;
    private int rear = 0;

    public boolean isEmpty() {
        return size == 0;
    }
}
```

The variable `front` is the position of first element in the queue, and `rear` points to the next available position in the array. Before an element is enqueued, `front` is equal to -1; after the first element is enqueued, `front` becomes non-negative, and never becomes negative again.

When adding an element to the queue, you add the element to position `rear`. Note that `rear` may eventually become equal to the size of the underlying array; at that time, the array needs to be resized before any enqueue operation is carried out. 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`.

If a user tries to peek or dequeue when the queue is empty, an `IllegalStateException` should be thrown, as in the following:

```
public int peek() {
    if(isEmpty()) throw new IllegalStateException("Cannot peek because
        Queue is empty!");
}
```

Notes:

1. Your queue must support unlimited `enqueue` operations. This means that you will need to incorporate a procedure for resizing the background array periodically.
2. You don't need to implement it as a circular queue. When the `rear` reaches the end of the array, you can start resizing the array.