Worksheet 7: Array-based Queues

In this worksheet you will implement an array-based version of the **Queue** interface. For this implementation, you should use the pseudo code provided in the lecture. The worksheet is intentionally the same as worksheet 6 so that you can confirm that your implementation works.

Do all the questions below (Q1 is worth 50%, Q2 and Q3 are worth 25% each). Submit 2 files: *ArrayQueue.java* and *ArrayQueueTest.java* (which contains the answer to Q3).

1. Create a class called **ArrayQueue** that implements the **Queue** interface provided. Include two constructors: a default constructor (no parameters) that creates a queue with a fixed capacity of 50; and a second constructor that takes a capacity as a parameter so that you can set the capacity of the queue when you create it. Implement the five methods specified in the **Queue** interface.

2. Override the toString() method to provide a string based representation of the state of the queue (use the toString() method on the **ArrayStack** class as inspiration). Try to make the output meet the following format:

   [3] : “A”, “P”, “E”, null, null, ...

   Where this represents a queue that contains 3 values A, P, and E, which were enqueued in that order.

3. Write a main method that performs the following operations on an integer queue (you must use the **ArrayQueue** implementation here):

   Enqueue(10), Enqueue(5), Dequeue(), Enqueue(15), Enqueue(3), Dequeue(), Enqueue(7), Dequeue(), Enqueue(20).

   Print out the state of the queue after each operation.

   Add a loop at the end that clears the queue and calculates the total value of the numbers were left in the queue at the end of the above sequence of operations.