Worksheet 8: Vectors

In this worksheet you will implement an array-based version of the **Vector** interface. For this implementation, you should use the pseudo code provided in the lecture.

Do all the questions below (Q1 is worth 70%, Q2 is worth 30%). Submit 2 files: *Vector.java* and *VectorQueue.java* (which contains the answer to Q3).

1. Create a class called **ArrayVector** that implements the **Vector** interface provided. Include two constructors: a default constructor (no parameters) that creates a vector 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 vector when you create it. Implement the size methods specified in the **Vector** interface (10% per method). For the remaining 10%, override the toString() method to provide a string based representation of the state of the vector (use the toString() method on the **ArrayStack** and **ArrayQueue** classes as inspiration). Try to make the output meet the following format:

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

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

   HINT: Implement this method before you implement the 4 main Vector methods (insertAtRank, removeAtRank, elementAtRank and replaceAtRank) and use it to check that each method implementation is working correctly.

2. Develop an implementation of the Queue interface called **VectorQueue** that uses your Vector implementation (20%) include a main method that tests your implementation by performing the following operations:

   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.