Exercise # 5 – Stack & Queue

You **MUST NOT** modify any existing codes in all given templates unless the instruction allows you to do so.

Part A: Stack

You are given interface Stack.

1. (6 marks) Class ArrayListStack is supposed to be an implementation of stack using Java's ArrayList. Complete class ArrayListStack from a given template:
   - Implement a default constructor and all methods from interface Stack.
   - Additional methods maybe required in order to run tests.
   - Use Class ArrayListStackTest for testing your ArrayListStack.

2. (6 marks) Class UseStack is given. Write code for method *sort(Stack s)*. This method receives a stack as its parameter, then **sort** the stack so that the smallest value is on top and the largest value is on the bottom:
   - You are allowed to use additional stacks to help, but you are not allowed to create other data structures such as arrays or lists.
   - You are allowed to create primitive variables.
   - Operations on stacks must only be operations defined in interface Stack.
   - Use class UseStackTest to test your code.

Part B: Queue

The interface MyQueue (with all method explanations) and class QueueArray are given.

1. (6 marks) In class QueueArray, write method reverse(). This method reverses order of data stored inside the queue. For example, if the queue originally stores 6,7,8,9 (6 being the front value), the method will change the order to 9,8,7,6. Use class QueueArrayTest to test your code.
2. (4 marks) A priority queue is a queue that is arranged such that the most important value is always retrieved first. In this exercise, the smallest value is the most important. Priority queue has the following methods:
   - push(int x): insert new data x into the priority queue. The data must be added so that the smallest data is retrieved first by method top() and pop().
   - pop(): remove the smallest data.
   - top(): return the smallest data.

   Implement these methods in the provided PriorityQueue class. You code **must not** assume any implementation of queue**. Use variable q as MyQueue only, do not cast it to another type.** You can test your code by using class PriorityQueueTest. **DO not use sort()**.