COS226 Week 2 Activity

1. *Generics, iterators, autoboxing, and mathematical analysis. Algorithms textbook 1.3*

   Use the class shown here: http://algs4.cs.princeton.edu/13stacks/ResizingArrayStack.java.html

```
ResizingArrayStack<Character> stack = new ResizingArrayStack<Character>();
stack.push('A');
stack.push('C');
stack.push('T');
for (char left : stack)
    for (char right : stack)
         StdOut.println(left + " " + right);
```

   (a) What does the above code fragment output to standard output?

   (b) If N characters are pushed onto the stack initially (instead of 3), how many lines of output does the above code fragment produce as a function of N?

2. *Unit testing.* Describe three ways to test the correctness of the implementation of `ResizingArrayStack`.

   -
   -
   -

3. *Stacks, queues, and amortized analysis. Algorithms textbook 1.3* Describe an implementation of a queue using two stacks, where each operation (*construct*, *push*, and *pop*) takes a constant *amortized* number of stack operations. Explain why, starting from an empty queue, any sequence of N queue operations takes proportional to N stack operations, in the worst case.

4. *Design an algorithm.* Design a quadratic-time algorithm for the 3-sum problem. Describe your design by giving a crisp and concise English description of your algorithm; don't write Java code.

   (a) Given an integer x and a sorted array a[] of N distinct integers, design a linear-time algorithm to find if there exists indices i and j such that `(a[i] + a[j] == x)`. Hint: start by checking whether `a[0] + a[N-1] is <, >, or == x`.

   (b) Given an array a[] of N distinct integers, design a quadratic-time algorithm to find if there exists indices i, j, and k such that `(a[i] + a[j] + a[k] == 0)`. Hint: Use the result from (a). You can assume the array is sorted since sorting the array can be done in quadratic (and even linearithmic) time.