CSC 212 Programming Assignment # 1
Recursive Postfix Expressions Evaluation
Due date: 18/12/2016

Guidelines:	This is an **individual** assignment. The assignment must be submitted to **Web-CAT**

In this assignment, do not use any auxiliary data structures (in particular, do not use a stack).

1. Write a recursive method, *eval*, to evaluate a postfix expression. The expression is represented as a String and contains the following operators: +, -, * and /. For simplicity, assume that all the numbers are single digit and unsigned, for instance 5, or 6 but not 23, 124 or -4. An example of an input is: "873-*4+23-*58-+".

   Programming hint: in order to transform a single character located at position i in a string *exp* to its numerical value, you may use:

   $$val = Character.getNumericValue(exp.charAt(i));$$

2. Write a recursive method, *infix*, to transform a postfix expression into an infix one. Use the same assumptions as in the previous question. For simplicity, put all operation between parentheses. For instance, the postfix expression "23+" is transformed to "(2+3)", and "873-*4+23-*58-+" is transformed to "((((8*(7-3))+4)*(2-3))+(5-8))".

```
class MyInt {
        public int val;
}

public class Postfix {
        // Public recursive method.
        public static double recEval(String exp, MyInt i) {

        }
```

```
        // Public non-recursive
        public static double eval(String exp) {
                MyInt i = new MyInt();
                i.val = exp.length() - 1;
                return recEval(exp, i);
        }

        // Public recursive method.
        public static String recInfix(String exp, MyInt i) {
        }

        // Public non-recursive
        public static String infix(String exp) {
                MyInt i = new MyInt();
                i.val = exp.length() - 1;
                return recInfix(exp, i);
        }
}
```