ECE 370 **Programming Assignment 2**
Winter 2016

Assigned 1/26/2016
Due 2/10/2016

In this program, you will write two functions:
1. INFIX_TO_POSTFIX -converts infix expression to postfix using stack and queue
2. EVALUATE_POSTFIX -evaluates postfix expression using a stack

You will also need PUSH, POP, and other procedures for manipulating stacks. You must implement stack and queue, as defined in the textbook or a style of similar. You **cannot** just make a function call to the standard stack or queue libraries. The stack and queue functions must be in your own code.

The infix expressions to be evaluated are follows. Your main program reads in an infix expression, calls INFIX_TO_POSTFIX to convert it to postfix form, and then calls EVALUATE_POSTFIX to evaluate it. For each infix expression, your program should print the original infix expression, its postfix expression, and the result of the evaluation (that is, the value of the expression). Your program should check for end-of-file and stop when there are no more infix expressions. After processing all the expressions, your program should print a final line that is the sum of all the values.

The only operators used in the infix expressions are multiplication (*), division (/), addition (+), subtraction (-), and exponential (^). Standard C++ precedence rules are observed. Parentheses are also used. As is customary, anything within parenthesis is evaluated before anything else is evaluated. You may assume there will be no unary minus. All operands are one-digit decimal numbers with no decimal point. The result of each calculation should be float.

The input data file name should be
"***a2.txt***"

Sample Test Data:

```
2^2
8-3*2
7+(4-6)*(8+6)/3
4+1+(2-1)
3/8+4*5
9*2+((4-3)*2)/2
```

Stack and queue examples can be found in the textbook. You may use arrays to implement stacks and queues, but they have to be defined with functionalities of stack and queue.