ECE 370 Fall 2012

Assigned 9/24/2012

Due 10/8/2012

Programming Assignment 2

In this program, you will need to

(1) write two functions:

INFIX_TO_POSTFIX -converts infix expression to postfix using stack and queue
EVALUATE_POSTFIX -evaluates postfix expression using a stack

(2) Write 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 expression, and then calls EVALUATE_POSTFIX to evaluate the postfix. For each infix expression, your program should print the original infix expression, the equivalent 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 resulted from postfix expressions.

In this assignment, the operators used in the infix expressions are multiplication (*), division (/), addition (+), subtraction (-), and exponential (^). Standard C/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:

```
8-3*2
7+(4-6)*(8+6)/3
4+1+(2-1)
0/1+4*5
9*2+((4-3)*2)/2
2^2
```

Stack and queue can be linked lists, or arrays. You may use arrays to implement stacks and queues, but they have to be defined with functionalities of stack and queue.