PROGRAMMING ASSIGNMENT # 4

Spring Semester **CSC 36000** **March 9, 2015**

PROGRAM STATEMENT: For your fourth programming assignment, you are to write a program that will take a mathematical expression in **INFIX** notation, convert it to **POSTFIX** notation, and then evaluate the **POSTFIX** notation form of the expression. The program is to employ one or more stacks with the EMPTY, PUSH and POP operations to complete the assignment. The program is due on **March 23, 2015.**

INPUT: Input for this program will consist of an unknown number of lines. Each line will contain a mathematical expression in INFIX notation. There will be only one expression per line of data. There will be no more than 30 characters per expression and all digit characters will be between 1 and 9 inclusive. (**NOTE: the input values are single digits but your program must account for results that are more then one digit in length.**) The only operators in the expression will
be **'+'** for add,**'-'** for subtract, '*' for multiply, and '/' for integer divide. Parentheses may also appear in the expression. You may assume all data lines are valid (i.e. All left parentheses have a matching right parenthesis.) An uppercase **X** as the first character in an expression will signify the end of the input data. The input file is **stack_in.txt (or stack_in_2008.txt or stack_in_2010.txt)**

PROCESSING: The program is to convert the INFIX expression to POSTFIX notation and then evaluate the POSTFIX expression.

OUTPUT: Output for this program is to consist of the information listed below. Output for each expression is to appear on a **separate page.** Output is to consist of 3 phases. They are the conversion phase, the evaluation phase, and the expression phase. Output for each phase is specified below. Alignment of output is to appear as shown in the example.

For the conversion phase:

The headings are: **CONVERSION DISPLAY**

Infix Expression **POSTFIX Expression** **Stack Contents (Top to Bottom)**

A running display of the contents of the **INFIX** buffer, the **POSTFIX** buffer, and the **stack.**

For the evaluation phase:

The headings are: **EVALUATION DISPLAY**

POSTFIX Expression **Stack Contents (Top to Bottom)**

A running display of the contents of the POSTFIX buffer and the stack

---

For the expression phase:

A. The heading is **ORIGINAL EXPRESSION AND THE ANSWER:**

The original expression as contained in the data file followed by the equal sign and the correct answer.

See page 2 for an example of the input data and the resulting output.

SAMPLE INPUT: Suppose the INPUT DATA looked like:

(3 + 4) * 6

SAMPLE OUTPUT: Then the program output should look like:

CONVERSION DISPLAY

Infix Expression	**POSTFIX Expression**	**Stack Contents (Top to Bottom)**
(3 + 4) * 6	Empty	Empty
3 + 4) * 6	Empty	(
+ 4) * 6	3	(
4) * 6	3	+ (
) * 6	3 4	+ (
* 6	3 4 +	Empty
6	3 4 +	*
Empty	3 4 + 6	*
Empty	3 4 + 6 *	Empty

EVALUATION DISPLAY

POSTFIX Expression	**Stack Contents (Top to Bottom)**
3 4 + 6 *	Empty
4 + 6 *	3
+ 6 *	4 3
6 *	7
*	6 7
Empty	42
Empty	Empty

ORIGINAL EXPRESSION AND THE ANSWER:

(3 + 4) * 6 = 42

NOTE: SPACING IS FOR READABILITY ONLY. THE ACTUAL DATA FILE(S) DO NOT CONTAIN SPACES IN THE EXPRESSIONS. YOU MAY PLACE SPACES IN YOUR OUTPUT AS SHOWN IF YOU SO DESIRE.