COMP 1603 Computer Programming III

2018/2019 Semester 2

Assignment 3

Given: Friday 22nd March, 2019.

Date Due: Friday 5th April, 2019, at 10:00 pm

No late assignments will be accepted.

Submission: Upload a zip file containing your **source code, input file and output file** to My Elearning by the deadline. Name your source file your first name initial + last name e.g. MSlater for Marcus Slater. Ensure that you place your name, student ID, course code, email address and assignment number in documentation at the top of your file.
In the documentation at the top of your source code, indicate approximately what percentage of your program worked. E.g. "My program worked 90%, but could not process one line of data properly".

Plagiarism form: Submit a plagiarism form for this assignment as a signed hard copy. Deposit your form in the COMP1603 box slot in the DCIT office. **(Note that all assignments are to be individually done)**

No My Elearning Access?
In this case you are to email your files (no .exe files) to COMP1603P3UWI@gmail.com.
Place the assignment number in the subject of the email.
You must indicate a reason why you have no My Elearning access in the email. The COMP1603P3UWI@gmail.com email address is not monitored otherwise.
(**Also** cc your email to Michael.Hosein@sta.uwi.edu).

Input/Output
All input data must be stored in a file **"input.txt".**
All output data must be stored in a file **"output.txt".**
All output must also be printed to the screen.

Description
An arithmetic expression is a string that can consist of **single digit** operands, parentheses, and the operators +, -, *, /, and ^. *x^n* denotes x^n where *n* is a non-negative integer.

Write a program **RPN.cpp** that repeatedly reads infix expressions from the input file and

1. Converts each expression to its postfix equivalent. Print the original expression followed by the postfix (Reverse Polish Notation) form.
   E.g.
   Infix: 3 + 4
   Postfix: 3 4 +

2. Evaluate each postfix expression from (1) and print the value of the evaluated expression. E.g.
   Postfix: 3 4 +
   Evaluation: 7

Input
Each line of the input file contains an arithmetic expression. One or more spaces may separate operands and operators. The last line in the data file contains a '$' only.

Sample data follows. Note that the marking may involve any input data file that matches the assignment description.
4*2^ (2*3)
4 * (8-6) ^ (9-5) - 8*2
4 * (7 – 6) / (9-5)) – 8*8
(8 + 6 * 5)/(8 - 3 * 2)
4 * (8 – 3) ^ (((7-5)) – 8*8
(9-7) * ((5 - 4 * 2)/3/5) * 2
(8/2/2)- 3^2
$

Division
Note that the '/' operator performs floating point division and **not** integer division
Therefore 5/2 = 2.5 (it is incorrect to calculate 5/2 as 2)

Error handling
There is only one possible error that can occur in an expression: unbalanced parentheses. Print a specific message if parentheses do not match.
Examples:
(1+2)) Error: more right than left parentheses
((1-5) Error: more left than right parentheses

If an expression has a parentheses mismatch, move on to the next expression after printing the error message. Do not attempt to convert to postfix or evaluate.
A stack must be used to check mismatched parentheses.

Note: Your program must not prompt for any input. A penalty may be incurred if the program prompts and waits for any information.

Queries: After the assignment is marked, queries can only be made on the exact files submitted. It is your responsibility to ensure that you submitted the correct files. Queries are to be made within one week of the release of the assignment results.