CSC 228 Data Structures and Algorithms, Spring 2020
Instructor: Dr. Natarajan Meghanathan

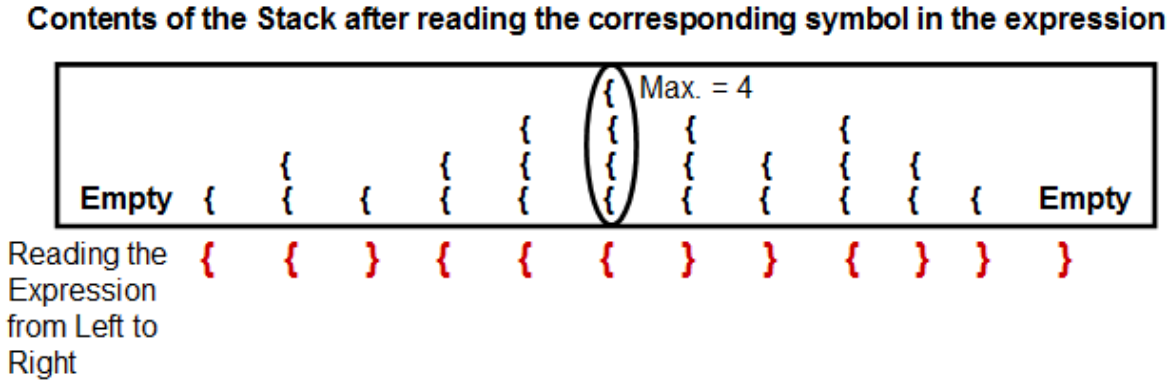
Assignment 4: Maximum Depth of Nested Parentheses in an Expression

Due by: Feb. 25th, 11.59 PM

Consider the code given to you to determine whether an expression of parentheses is balanced or not using a Doubly Linked List implementation of a Stack. Modify the code (given in the main function) to determine the maximum depth of nested parentheses in a given expression, provided the expression is balanced.

Note that your code should be also able to check whether a given expression of parentheses is balanced or not. If an expression is not balanced, you should not print the value for the maximum depth of nested parentheses: your program should just print the message that the expression is not balanced and terminate.

For simplicity, you can assume that the only parenthesis symbols of use are { and }. The maximum depth of nested parentheses in a balanced expression is the largest number of open parentheses { in the stack at any time. For example, the maximum depth of nested parentheses in the expression { { } { { { } } { } } } is 4.



Run your modified code with the following expressions and determine the maximum depth of nested parentheses of each, if the expression is balanced. Include screenshot displaying the result.

a. { { } { { { } } { } } }
b. { { { } } { { { } } } }
c. { { { } { { } { { } } } { } } }
d. { { { } } { { } } }
e. { } { { } { } }

Submission

Submit items (1) and (2) together as one PDF file
(1 - 25 pts) Briefly describe your algorithm (along with a pseudo code) to determine the maximum depth of nested parentheses in a given expression. Analyze the time complexity of the algorithm.
(2 - 10 pts) Screenshots of the execution of the algorithm/code for each of the above five inputs (a)-(e).

(3 - 65 pts) **Submit a separate C++ file** that has the complete code (including the modification/implementation of the algorithm in the main function)