COSC 2436 - Group Assignment 2

Introduction: This group assignment will utilize stacks and queues to determine whether expressions are valid and then print out the expressions in the order in which they appeared in the input file.

Determining whether expressions are valid or invalid: There are three components of the expression you will be checking to determine whether an expression is valid or invalid:

- Valid Parentheses - An expression is considered to have valid parenthesis if the open parenthesis matches the closing parenthesis.

  Valid Parentheses:

  (4+2)-[1*3]

  {5+4-[3*(2-2)]/3}

  Invalid Parentheses:

  ]4+{1-2(()

  [3+1)+{2*8)

- Double Parenthesis - An expression is considered to have double parentheses if the expression contains back-to-back parenthesis.

  Double Brackets:

  ((4+2))

  {5+{[3-1]}*2}

  Not Double Brackets:

  (1)+(1)+(1)

  {1+[2-(3)]*0}

- Correct Order of Parentheses - Parenthesis are in the correct order if curly brackets are outside of curly brackets, square brackets, and/or parenthesis, square brackets are outside of square brackets and parentheses, and parenthesis are outside of parentheses.

  Correct Order:

  {5+[2-(3-2)+1]/3}

  Incorrect Order:

  [5+(2-{3-2}+1)/3]

  (1+[5+2]*(7+9))

An expression is considered valid if it has all of the following attributes: valid parentheses, no double parenthesis, correct order.

An expression is considered invalid if it has one or more of the following attributes: invalid parentheses, double parenthesis, incorrect order.

Assumptions:

- The only parenthesis that you will see will be:
    - Curly Brackets { }
    - Square Brackets [ ]
    - Parenthesis ( )
- The numbers in each expression will only be single digit, nonnegative numbers (0 through 9)
- The only operators will be: +, -, *, and /
- Input will never be empty
- There will be no empty lines in input
- There will be no blank spaces in input. This means nothing like: {4 + [9   - 1] /2   +1 }
- You may use the C++ STL stack and queue

Output: You should output the expressions based on whether they are valid or invalid and also keep the order in which they appeared in the input file. An example of output is shown below.

input1.txt

```
1+2+3+4+5+6
(1+2+3+4+5+6)
[1+2+(3+4)+5+6]
{1+[2+(3+4)+5]+6}
{1+[2+(3+4}+5)+6]
]1+2+]3+4]+5+6]
((1+2+3+4+5+6))
(1+[2+{3+4}+5]+6)
```

output1.txt

```
Valid
1+2+3+4+5+6
(1+2+3+4+5+6)
[1+2+(3+4)+5+6]
{1+[2+(3+4)+5]+6}
Invalid
{1+[2+(3+4}+5)+6]
]1+2+]3+4]+5+6]
((1+2+3+4+5+6))
(1+[2+{3+4}+5]+6)
```

Directions for submitting group assignment 2:

We expect every student of each group will participate to solve the problem and discuss with each other. The purpose of this group assignment is to learn about stacks and queues.

Every student of each group needs to submit the same copy of the solution. Group assignment 2 needs to be turned into the server for this class. Make sure to create a folder under your root directory, name it **ga2** (must be in lower case). Only copy your .cpp files, .h files, and ArgumentManager.h file to your **ga2** folder. This assignment will only be graded once, so make sure you are submitting the correct solution before the due date because you will not be able to resubmit the assignment.

Due Date: Tuesday, March 21, 2023 by 11:59 pm