COSC 2436 Group Assignment 2

Introduction:

You and your group will implement a C++ program to find valid expressions, get their priority, input them into a priority queue based on the expression's priority, and then print the expressions in order of their priority. The purpose of this group assignment is to get students familiar with stack and priority queue operations. Please name the folder on the server as **ga2**.

Input Files:

- Each line of input will either contain an expression or an empty line
- Empty lines and invalid expressions should be ignored.
- An expression is invalid if it has redundant brackets/parenthesis and/or it has invalid brackets/ parenthesis.
- Here are a few examples of invalid expressions:
    - **{ 1 + 3 + (5 * 4)]** *Here the expression is invalid because it begins with a* **{** *but ends with a* **]**
    - **((4 + 9 * 7 - 2))** *Here the expression is invalid because the* **( )** *are redundant*
    - **[1 + 2 - (9 * 0)) / 3 [** *Here the expression is invalid because it has an extra* **)** *and it ends with a* **[** *and not a* **]**
- Here are a few examples of valid expressions:
    - **{ 1 + [3 * (4 / 2)] - 1 }**
    - **(4 - [2 + { 0 * 0 } - 1] + 1)**
    - **(1) + (2) - [3] + {0}**
- If the expression is valid, it should be added to a priority queue based on its priority.
- The priority of the expression will be the answer of the expression.
- For example, the expression **{1+(2*2)+9}** will have a priority of **14** because **{1+(2*2)+9} = 14**.
- For this assignment, <u>**LOWER ANSWER MEANS HIGHER PRIORITY!**</u>

- If two expressions have the same priority, they should be inserted into the priority queue in the order in which they appeared in input. This is in keeping with the “First In, First Out” quality of a queue.
- For example, if your input reads in the expression **(1+3)** and then reads in the expression **(5-1)**, the expression **(1+3)** will come before the expression **(5-1)** because **(1+3)** appeared before **(5-1)** in the input file.

Output:

- Output file should display every expression in your priority queue
- If there are no valid expressions, print ***No Valid Expression*** to the output file.

Assumptions:

- Each expression will have at least two numbers and one operator
- The numbers in the equation will always be a single-digit integer ($0 \leq x \leq 9$)
- The operators will be addition (+), subtraction (**-**), multiplication (*****), and/or division (/)
- The priority will always be an integer between -1000 and 1000 ($-1000 \leq priority \leq 1000$)
- There will be no spaces in the expressions
- There will be between 0 and 1000 expressions for each input file ($0 \leq n \leq 1000$)

Turning in ga2:

<u>Every member in the group must submit the same solution.</u> The lowest grade of a group will be counted for every group member, so make sure everyone submits the same copy.

When submitting ga2 to the server, include your main.cpp file, Argument Manager.h file, and any other .h files you used when creating your solution. Name the folder for this assignment **ga2** on the server. After your ga2 folder is uploaded to the server, run the command “***chmod -R** 777 **ga2***" in your root directory to make sure that we have the permissions necessary to test your code.

This assignment will only be graded **<u>once</u>**. There will be no late submissions or regrades for this assignment.

The due date for this assignment is **Tuesday, November 8, 2022 by 11:59pm**.

Input 1:

****NOTE**: Here, I’ve indicated which expressions are valid/invalid but know that your input file will not tell you this

```
{[1+(9/(2+1)+8)]*(1+1)}    Valid

[3+(4)*0+(3/3)]    Valid

{(9+2-[8*7]+4)}    Invalid

{12*2-(20*1)+0-[2*1]+2}    Valid
{12*2-(20*1))+0-[2*1]+2}    Invalid
(3*9/(4-1)+[9+8})

1+1  Valid

(4+9+[8*2]-1)    Valid
```

Answer 1:

```
{12*2-(20*1)+0-[2*1]+2}
1+1
[3+(4)*0+(3/3)]
{[1+(9/(2+1)+8)]*(1+1)}
(4+9+[8*2]-1)
```

**Note: This document may have typos, if you think something is illogical, please email the TAs for confirmation.