Group Assignment 2: Stacks & Queues

The purpose of this group assignment is to get familiar with using stacks and queues to solve problems. In this assignment you will write a program to detect valid and invalid expressions. Each expression will consist of different types of brackets: **{**, **}**, **(**, **)**, **[**, and **]**. These will be the only characters in the expression. There will be no letters, no numbers, no spaces, no empty lines, and no other symbols except for these, although you should remember to remove the newline ('**\n**') and return ('**\r**') characters when you read in the expression from the input file. The input file will never be empty.

An expression is valid if:

- there is a corresponding closing bracket for every open bracket
- there is a corresponding open bracket for every closing bracket
- the closing bracket matches the open bracket
- the brackets are in the correct order

Brackets are in the correct order if:

- For a set of corresponding brackets, the open bracket is before the closed bracket
- Curly Brackets **{ }** are only enclosed in other curly brackets
- Square Brackets **[ ]** are only enclosed in either curly or square brackets
- Round Brackets **( )** are enclosed in either curly or square or round brackets
- Correct Order: **{ [( )] }**
- Incorrect Order: **([{ }])**
- Incorrect Order: **}]) ([{**

You will categorize each expression as either valid or invalid. You will then print each expression to the output file. You will print invalid expressions before valid expressions. You should print "Invalid" before printing the invalid expressions and "Valid" before printing the valid expressions. If there are no invalid expressions do not print "Invalid" and if there are no valid expressions, do not print "Valid". The expressions should be printed in the order they appeared in the input file. An example is given below to show you the input file with the corresponding output file.

You should submit your code to the server for grading. You must name your folder **ga2** and it should be located in your root directory. You should only include your main.cpp file, ArgumentManager.h file, and any other .h/.cpp files which are part of your solution in your ga2 folder. Everyone in the group must work together on this assignment and everyone in the group must submit the solution to the server. If you do not submit your solution to the server, you will receive a 0 for ga2. This will only be graded once.

You must use stacks and/or queues in this assignment. You are allowed to use the C++ stack and queue libraries.

input1.txt

```
{[()]}
{}
[()]
}[]}
(())
[{()}]
(][}}
}}}
```

output1.txt

```
Invalid
}[]}
[{()}]
(][}}
}}}
Valid
{[()]}
{}
[()]
(())
```

Check Canvas to see when this assignment is due.