Kyla Wilson
Feb 25, 2020
J00813814
Assignment 4

1.) For my implementation of finding maximum depth of Parentheses I create a total of 2 variables, “maxDepth” keep the count of the maxium number of parenthesis and “depth” keep the count of parentheses total. Whenever an open parentheses is found, “depth” variable is increment by one. After that statement, I check if the depth is greater than maxDepth; If so, I change the “max depth” to be equal to the “depth”. On the other hand, whenever a close parenthesis is found, I decrement the “depth” by one.

```
While( something in the stack ) {

        If( symbol == "{" ) {

                Depth++;

                If( depth > maxDepth ) maxDepth = depth

        } else if( symbol == "}" ) {
                if( top == "{" && symbol == "}" ) {

                        Depth--;

                }

        }

}
```

My algorithm has only two cases.

Worst case:

The expression is balanced, that way it will loop to end of expression.

Best case:

The expression is empty or it is not balanced, that way it will not loop through entire expression.

2.)

```
Enter an expression: {}{{}{}
Expression not balanced!!
Depth: 2
```

```
Enter an expression: {{{}}{{}}}
{{{}}{{}}} is balanced!!
Depth: 3
```

```
Enter an expression: {{{}{{}{{}}}{}}}
{{{}{{}{{}}}{}}} is balanced!!
Depth: 5
```

```
Enter an expression: {{{}}{{{}}}}
{{{}}{{{}}}} is balanced!!
Depth: 4
```

```
Enter an expression: {{}{{{}}{}}}
{{}{{{}}{}}} is balanced!!
Depth: 4
```