

Assignment 1 Report: Mathematical Parenthesis Checker

COMP 201 Data Structures and Algorithms

Name Surname0 :Ramazan Yetişmiş
Student ID : 041701013
Date : November 11, 2018

1. Algorithm Explanation

In this assignment, I stored parenthesis which is given by the user into the stack. Firstly, it took input from the user as a String line. Then turned the line into a char array. After that added some conditions to push the parenthesis into stack correctly. By looking my conditions, it can simply be said that, first condition checks whether the character is an opening parenthesis if so, I pushed it into the stack, if not I looked the stack If it had no elements in it and the parenthesis is a closing type, then the I returned false. This loop goes through The char arrays length. Finally, if the opening and closing parenthesis are matched, then the element popped From the Stack and at last I looked the stack to see whether it was full or empty

2. Outputs

In Figure 1 shown the math expression's, that was taken from the user, outputs. I used the following configurations in these plots. In the first plot there are first 6 output which is given us from the pdf file the output statements include all the conditions such as {"Parentheses are correct"," Parentheses do not match: Final stack is not empty!"," Parentheses do not match"}, In the second plot there are 5 outputs which I gave To see if it works for the other statements and by looking the outputs, we can say that algorithm works as it has to be.

FIGURE 1

```
Enter a mathematical expression: (4-3)*[8/3]
Stack Contents: [top]    (    :[bottom]
Stack Contents: [top]    [    :[bottom]
Parentheses are correct.

Enter a mathematical expression: [[5/2]*[(2-3)*4]]/(2*8)
Stack Contents: [top]    [    :[bottom]
Stack Contents: [top]    [ [    :[bottom]
Stack Contents: [top]    [ [    :[bottom]
Stack Contents: [top]    ( [ [    :[bottom]
Stack Contents: [top]    (    :[bottom]
Parentheses are correct.

Enter a mathematical expression: ))2+1((
Parentheses do not match!

Enter a mathematical expression: [2(1)4)
Stack Contents: [top]    [    :[bottom]
Stack Contents: [top]    ( [    :[bottom]
Parentheses do not match!

Enter a mathematical expression: [[5/2]*[(2-3)*4]]/(2*8)+[[[7]]
Stack Contents: [top]    [    :[bottom]
Stack Contents: [top]    [ [    :[bottom]
Stack Contents: [top]    [ [    :[bottom]
Stack Contents: [top]    ( [ [    :[bottom]
Stack Contents: [top]    (    :[bottom]
Stack Contents: [top]    [    :[bottom]
Stack Contents: [top]    [ [    :[bottom]
Stack Contents: [top]    [ [ [    :[bottom]

Parentheses do not match: Final stack is not empty!

Enter a mathematical expression: [[[]]]
Stack Contents: [top]    [    :[bottom]
Stack Contents: [top]    [ [    :[bottom]
Stack Contents: [top]    [ [ [    :[bottom]
Parentheses are correct.
```

PLOT 1

Enter a mathematical expression: $[(5+)-2$
Parentheses do not match!

Enter a mathematical expression: $(5*8)[2+8)]54*8)(($
Stack Contents: [top] (:[bottom]
Stack Contents: [top] [:[bottom]
Parentheses do not match!

Enter a mathematical expression: $[[[[]] [57+87*(85)*[4/7]]$
Stack Contents: [top] [:[bottom]
Stack Contents: [top] [[:[bottom]
Stack Contents: [top] [[[:[bottom]
Stack Contents: [top] [[:[bottom]
Stack Contents: [top] ([[:[bottom]
Stack Contents: [top] [[[:[bottom]

Parentheses do not match: Final stack is not empty!

Enter a mathematical expression: $[[[[[]]]][[(())()]]]$
Stack Contents: [top] [:[bottom]
Stack Contents: [top] [:[bottom]
Stack Contents: [top] [[:[bottom]
Stack Contents: [top] [[[:[bottom]
Stack Contents: [top] [:[bottom]
Stack Contents: [top] [[:[bottom]
Stack Contents: [top] ([[:[bottom]
Stack Contents: [top] (([[:[bottom]
Stack Contents: [top] ((([[:[bottom]
Parentheses are correct.

Enter a mathematical expression: $[2(1]4)+56*(88/7)$
Stack Contents: [top] [:[bottom]
Stack Contents: [top] ([:[bottom]
Stack Contents: [top] (:[bottom]
Parentheses are correct.

PLOT 2