

1- My Main Code's Algorithm

- 1 – I import scanner class for read the inputs from given by keyboard.
- 2 – I declare and initialize a stack object from myStack class for store and match parenthesis.
- 3 – I store the inputs in a string which are from scanner and using toCharArray method, I convert strings to char array.
- 4 – I create a for loop for control the parenthesis matching.
- 5 – I use try-catch for handling underflow errors and controlling the parenthesis matches. If the stack is empty I throw an exception.
- 6 – In if conditions I control the lefthand brackets and push them, I use else if conditions for when I pop an element, match the returned character did they match or not?

2- The myStack Class Algorithm

- 1 – The class has already given for me from my instructor.
- 2 – The class contains Stack features which are data fields, Stack's operations, methods and some conditions. I used them in the main function.

3- Sample Outputs

```
((([23-14[(35*11)])]*23+25))
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's last status: There is such an element in the stack
```

There is/are some Parantheses in the stack

```
[[53*23]]([([23-123]))]
Stack Contents: [Top] [ [bottom]
Stack Contents: [Top] [ [ [bottom]
Stack Contents: [Top] ( [bottom]
Stack Contents: [Top] [ ( [bottom]
Stack Contents: [Top] ( [ ( [bottom]
Stack Contents: [Top] [ ( [ ( [bottom]

Parantheses are correct!
```

Parantheses are correct

```

Enter a mathematical expression:
((32*34))[(21-23)][[3]2]
Stack Contents: [Top] ( [bottom]
Stack Contents: [Top] ( ( [bottom]
Stack Contents: [Top] [ [bottom]
Stack Contents: [Top] ( [ [bottom]
Stack Contents: [Top] [ [bottom]
Stack Contents: [Top] [ [ [bottom]

Parantheses are correct!

```

Parantheses are correct

```

Enter a mathematical expression:
3/(12-23(23)(23(2313))
Stack Contents: [Top] ( [bottom]
Stack Contents: [Top] ( ( [bottom]
Stack Contents: [Top] ( ( ( [bottom]
Stack Contents: [Top] [ ( ( [bottom]
Stack Contents: [Top] ( [ ( ( [bottom]

Stack's last status: There is such an element in the stack

```

There is/are some Parantheses in the stack

```

Enter a mathematical expression:
[/((23-34)][23-23(23*23*0)]]]
Stack Contents: [Top] [ [bottom]
Stack Contents: [Top] [ [ [bottom]
Stack Contents: [Top] ( [ [ [bottom]
Stack Contents: [Top] [ [ [bottom]
Stack Contents: [Top] ( [ [ [bottom]

Parantheses do not match!

```

Parantheses do not match

```

Enter a mathematical expression:
3))123(23)[213](231)))
Parantheses do not match!

```

Parantheses do not match