Parenthesis Matching Problem using Stack Data Structure :

→ What is parenthesis matching ?

→ We have to see if the given expression has balanced brackets which means every opening bracket must have a corresponding closing bracket and vice versa.

((3 * 2) - 1 (8 - 2))

Balanced Parantheses.

~~3~~ 1 - 3) * 4 (8

NO CORRESPONDING OPENING BRACKET

NO CORRESPONDING CLOSING BRACKET

Unbalanced Parantheses.

→ Checking if the parantheses are balanced or not must be a cakewalk for humans since we have been dealing with this for the whole time. But even we

would fail if the expression becomes too large with a great number of parantheses. This is where automating the process helps.
And for automation, we need a proper working algorithm.

→ We'll use stacks to match these parantheses:

1) Assume the expression given to you as a character array.

3 * 2 - (8 + 1) =>

0	1	2	3	4	5	6	7	8	9
3	*	2	-	(	8	+	1	)	\0

2) Iterate through the character array and ignore everything you find other than the opening and the closing parentheses. Every time you find an opening parenthesis, push it inside a character stack. And every time you find a closing parenthesis, pop from the stack, in which you pushed the opening bracket.

3) Conditions for unbalanced parantheses:

→ ⊗ When you find a closing parenthesis and try achieving the pop operation in the stack, the stack must not become underflow. To match the existing closing parenthesis, at least one

opening bracket should be available to pop. If there is no opening bracket inside the stack to pop, we say the expression has unbalanced parantheses.

→ For ex: the expression (2+3)*6)1+5 has no opening bracket corresponding to the last closing bracket. Hence unbalanced.

→ At EOE, that is, when you reach the end of the expression, and there is still one or more opening bracket left in the stack, and it is not empty, we call these parantheses unbalanced

→ For ex: the expression (2+3)*6(1+5 has 1 opening bracket left in the stack even after reaching the EOE. Hence unbalanced.

4) NOTE: Counting and matching the opening and closing brackets numbers is not enough to conclude if the parenthesis are balanced. For eg: 1+3)*6(6+2

EXAMPLE:

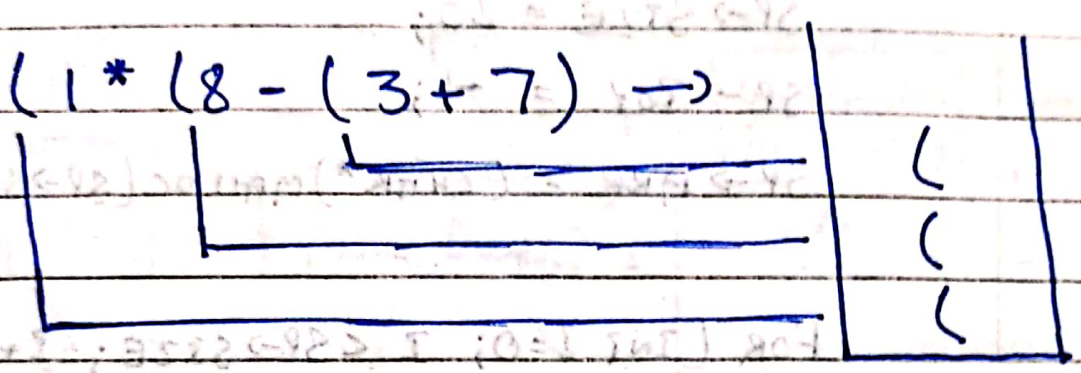
(1*(8-(3+7) →

0	1	2	3	4	5	6	7	8	9	10
(	1	*	(	8	-	(	3	+	7	)

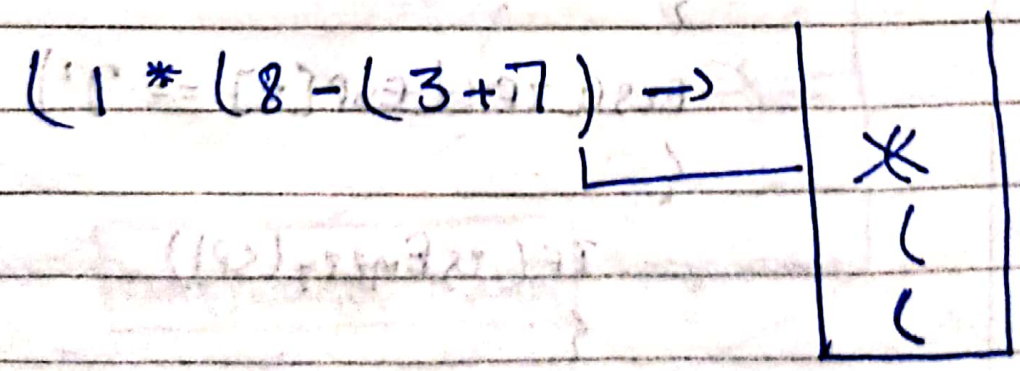
→ We'll try checking if these above

expression has balanced parantheses or not.

→ STEP 1: Iterate through the char Array, and push the opening brackets at positions 0, 3, 6 inside the stack.



→ STEP 2: Try popping an opening bracket from the stack when you encounter a closing bracket in the expression.



→ STEP 3: Since we reached the EOE and there are still two paranthesis left in the stack, we declare this expression of paranthesis unbalanced.