* Data Structure Lab (DSL) - Practical Number - 10 (Group - D)

Name :- Kaustubh Shrikant Kabra.
Class :- Second Year Engineering
Div :- A Roll Number :-
Batch :-
Department :- Computer Department
College :- AISSMS's IOIT.

Title :-

Write a C++ program using stack to check whether given expression is well parenthesized or not.

Aim :-

In any language program, mostly syntax error occurs due to unbalancing delimiter such as (), {}, []. Write a C++ program using stack to check whether given expression is well parenthesized or not.

Objective :-

1) To study the stack data structure.
2) To study the operations on stack.

Theory :-

A stack is an ordered list in which all the insertions and deletions are made at one end. It possesses the property of LIFO i.e. last in first out.

Stack Operations :-

1) Basically there are two important stack operations a) Push
b) Pop.

2) Performing push operation means we are inserting the element onto the stack. While, pop operation means we are removing the element from the stack.

3) Before pushing, we need to check stack full condition and before performing pop operation we need to check stack empty condition.

Algorithm :-

Step 1 - Start

Step 2 - Declare a class to create a stack, constructor and methods to perform operations on stack.

Step 3 - Accept an expression from the user.

Step 4 - Traverse to the expression and push the opening parentheses in stack.

Step 5 - If no parenthesis are present i.e. stack is empty, then display well balanced and stop.

Step 6 - Pop an element from the stack if for every opening parenthesis ('(', '[', '{'), there is a corresponding closing parenthesis

Step 7 - Repeat step 6 until stack becomes empty.

Step 8 - If stack becomes empty i.e. there is a closing parethesis for every opening parenthesis, return true.

Step 9 - If the return value is true, then the equation is well balanced.

Step 10 - Go to step 3, if user wants to check another expression.

Step 11 - Stop.

Analysis:-

Time complexity -

1) Display → O(n)
2) Push and pop → O(1)
3) Check parenthesis → O(n).

Conclusion:-

Hence, we have checked whether a given function's expression is well parenthesized or not.