

Assignment-g



Name: Shrikrushna S Zinde

R. No : 21286

Prb statement:-

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

Objective:-

- To understand Concept of stack data structure.
- To study diff. operation associated with stack data structure.

Outcome:-

- To write Function for stack operation.
- Determine syntax based on stack quantity.

SW packages and H/W app used:-

Operating system (64-bit) 64 bit
Fedora 17. programming tools
(64 bit) latest open Source
update of Eclipse programming
Framework, C++

Theory:-

Stack:-

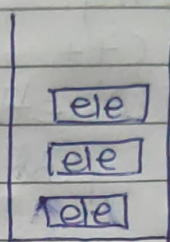
Stack is a linear data-structure which follows a particular order in which the operations are performed.

Stack is based on LIFO (last in first out). The element which is added last is removed first.

Push LIFO



POP



Stack

Operations:-

Push () : pushing / adding the element in the stack.

Pop () : removing / deleting the element from stack.

Application:-

- Reverse a word
- Infix, prefix, postfix Conversion.
- Function Call handling
- Backtracking.

Algorithm:-

1. declare a stack of 18 character.
2. now traverse through given expression string.
3. if character is opening bracket, (, [, { push it in the stack.
4. if character is closing bracket, then pop the character from stack. if both characters are matching then traverse else unbalanced string.
5. After complete traversal if the stack is non-empty then string is not well parenthesized.

Complexity:- $O(n)$

Test Case.	Describe if	exp. output.	out. put.	Status.
1	check. string (ab[c])	Well parenthesized.	Well parenthesized	Pass.
2	{[] }	Not parenthesized	Not parenthesized	Pass

Conclusion:- Successfully implemented the prog. using stack & S