GROUP - D

Practical No: 11 (D-26)

Title: Write C++ program to check well formedness of parenthesis using stack.

Objectives :

- To accept expression from user
- To use stack to perform operations.
- To check whether the given expression is well parenthesized or not.

Problem 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 parenthesized or not.

Outcome :

- Display Expression accepted by user
- Result of checking well formedness of parenthesis.

Operating System recommended :- 64-bit Open source Linux or its derivative

Programming tools recommended :- Open Source Python, Programming tool like Jupyter Notebook, Pycharm, Spyder, G++/GCC

Hardware Requirements :

i3 or above processor , 2 GB or above RAM, 512 GB or above Hard-disk etc

Reference for theory : https://www.javatpoint.com/data-structure-stack

Theory :

- **What is Stack ? Explain working of stack , Advantages, Disadvantages, Applications?**
- **Explain Stack as an ADT(operation of stack)?**
- **Explain Check for Balanced Bracket expression using Stack.**
- **Difference between Stack and Queue.**

NOTE : just write heading and try to explain in details with examples. Some websites are provided as reference(don't write reference). You can use other website also.

Write algorithm/pseudo code for checking wellformed parenthesis :

a) To check empty stack

b) To check full stack

c) To initialize stack

d) To Push elements to the stack

e) To pop element from stack

f) To display stack as wellformed parenthesis expression

Algorithm:

Write Algorithms for program/code which you have implemented.

Flowchart :

Draw flowchart for above algorithm

Conclusion:

Thus, We have successfully checked well formedness of parenthesis using stack

Continuous Assessment of Student:

TS	PR	UC	VA	RN	Total Marks	Faculty Signature
(2)	(2)	(2)	(2)	(2)	(10)	

- TS – Timely Submitted, PR- Performance, UC- Understanding of Code, VA- Viva Answered, RN- Regularity and Neatness

Practical No: 12 (D-27)

Title: Write a C++ program for expression conversion as **infix to postfix** and its evaluation using stack.

Objectives :

- To accept input expression from user
- To convert infix to postfix expression.
- To display output as postfix expression

Problem Statement: - Implement C++ program for expression conversion as infix to postfix and its evaluation using stack based on given conditions:

1. Operands and operator, both must be single character.
2. Input Postfix expression must be in a desired format.
3. Only '+', '-', '*' and '/ ' operators are expected.

Outcome :

- Display infix Expression accepted by user
- Equivalent postfix expression
- Result of evaluation of an expression.

Operating System recommended :- 64-bit Open source Linux or its derivative

Programming tools recommended :- Open Source Python, Programming tool like Jupyter Notebook, Pycharm, Spyder, G++/GCC

Hardware Requirements :

i3 or above processor , 2 GB or above RAM, 512 GB or above Hard-disk etc

Reference for theory : https://www.javatpoint.com/data-structure-stack

Theory :

- **What is polish notation? List different notations with examples. Mention operator precedence in tabular form.**
- **Explain Infix expression. Postfix expression with examples.**
- **Explain algorithms to convert infix to postfix expression with example.**
- **Write algorithms for postfix expression evaluation with example.**

NOTE : just write heading and try to explain in details with examples. Some websites are provided as reference(don't write reference). You can use other website also.

Algorithm:

Write Algorithms for program/code which you have implemented.

Flowchart :

Draw flowchart for above algorithm

Conclusion:

Thus, We have successfully performed conversion of infix to postfix expression using stack

Continuous Assessment of Student:

TS	PR	UC	VA	RN	Total Marks	Faculty Signature
(2)	(2)	(2)	(2)	(2)	(10)	

- TS – Timely Submitted, PR- Performance, UC- Understanding of Code, VA- Viva Answered, RN- Regularity and Neatness