S.Y. B. Tech. Academic Year 2022_23 Trimester: III

Fundamentals of Data Structures

LABORATORY WRITE UP

Experiment Number: 07

TITLE: Expression conversion using stack

PROBLEM STATEMENT:

Implement stack as an ADT and apply it for different expression conversions (infix to postfix or infix to prefix (Any one), prefix to postfix or prefix to infix, postfix to infix or postfix to prefix (Any one)).

OBJECTIVE:

1. To study Stack and its operations
2. To study the importance of expression conversions

THEORY: *//To be Written by Students*

// Write theory by elaborating below points

Write in brief about stack and different Expression conversion

IMPLEMENTATION:

- **PLATFORM:**
    - 64-bit Open source Linux or its derivatives.
    - Open Source C Programming tool like gcc/Eclipse Editor.

- **TEST CONDITIONS:-**

1. Input infix expression and check the postfix or prefix output.

2. Input postfix or prefix expression and check its infix , prefix or postfix.

- **PSEUDO CODE:**   *//To be Written by Students*

  Write pseudo code for push, pop, infix to postfix or infix to prefix(any one), prefix to infix , prefix to postfix, postfix to infix , postfix to prefix(any one which you implemented) operation.

- **TIME COMPLEXITY:**   *//To be Written by Students*

  Write time complexity of above operations

- **CONCLUSION:**

  Thus, implemented Stack Operations assignment using Array concepts. This System is able to apply different expression conversions.

- **FAQs**   *//To be Written by Students*

  - What is the advantage of prefix and postfix over infix expression?
  - Explain how postfix/prefix expression is evaluated.
  - What is ISP and ICP?
  - Give various applications of stack
  - Why is stack used in expression conversion?
  - Give stack full and stack empty conditions for stack.

- **PRACTICE ASSIGNMENTS**
  1. Write a program for evaluation of postfix expressions.
  2. Write a program to check whether a string is a palindrome or not using stack.
  3. Write a program to convert decimal to binary using stack.