National University of Computer and Emerging Sciences



Lab Manual 04

Data Structures Lab

|  |  |
| --- | --- |
| Course Instructor | Mr. Sarim Baig |
| Lab Instructor (s) | Hamna Waseem  Saif Ali |
| Section | F |
| Semester | Spring 2021 |

Department of Computer Science

FAST-NU, Lahore, Pakistan

## Objectives

After performing this lab, students shall be able to:

* Applications of Stacks
* Expression Evaluation using Stack.

Note-> You will use stack class implemented in Lab1.

**TASK 1:**

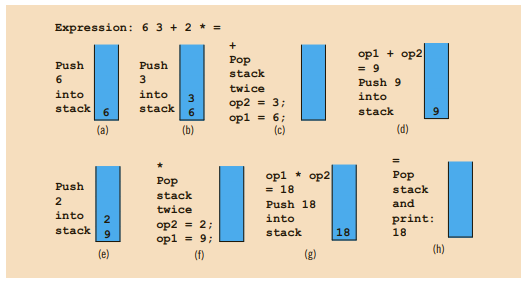
Write a function that evaluate a postfix expression using stack. You are given pseudo code to help you.

**Sample input:** 63+2\*=

**Sample Output:** 18

**Algorithm:**

* Create a stack to store operands (or values).
* Scan the given expression and do following for every scanned element.
  + If the element is a number, push it into the stack.
  + If the element is a operator, pop operands for the operator from stack. Evaluate the operator and push the result back to the stack.
* When the expression is ended (=), the number in the stack is the final answer.



**TASK 2:**

Write a function that evaluate a prefix expression using stack. You are given pseudo code to help you.

**Sample input:**  =\*+69 -31

**Sample Output:** 30

**Algorithm:**

Start from the last element of the expression.

* check the current element.
  + if it is an operand, push it to the stack.
  + If it is an operator, pop two operands from the stack. Perform the operation and result push back to the stack.
* Do this till all the elements of the expression are traversed and return the top of stack which will be the result of the operation.

**TASK 3:**

Write a function that takes an Infix expression convert it to a postfix expression using stack and then return it.

**Sample input:** 1+2\*3

**Sample Output:** 123\*+

**Algorithm:**

1. Read in the tokens one at a time
2. If a token is an integer, write it into the output
3. If a token is an operator, push it to the stack, if the stack is empty. If the stack is not empty, you pop entries with higher or equal priority and only then you push that token to the stack.
4. When you finish reading the string, you pop up all tokens which are left there.
5. Arithmetic precedence is in increasing order: '+', '-', '\*', '/';