EXPERIMENT NO:-03

Date of Performance:

Date of Submission:

Aim : Evaluate Postfix Expression using stack ADT.

Theory:

An infix expression in a High Level Language program is converted into its postfix form on its compilation time, since the evaluation of a postfix expression is much simpler than direct evaluation of an infix expression. The postfix expression is evaluated using Stack. Following is the method for evaluation postfix expressions:

1) Create a stack to store operands.

2) Scan the given expression and do following for every scanned element.

a) If the element is a number, push it into the stack.

b) If the element is an operator, pop operands for the operator from stack.
Evaluate the operator and push the result back to the stack.

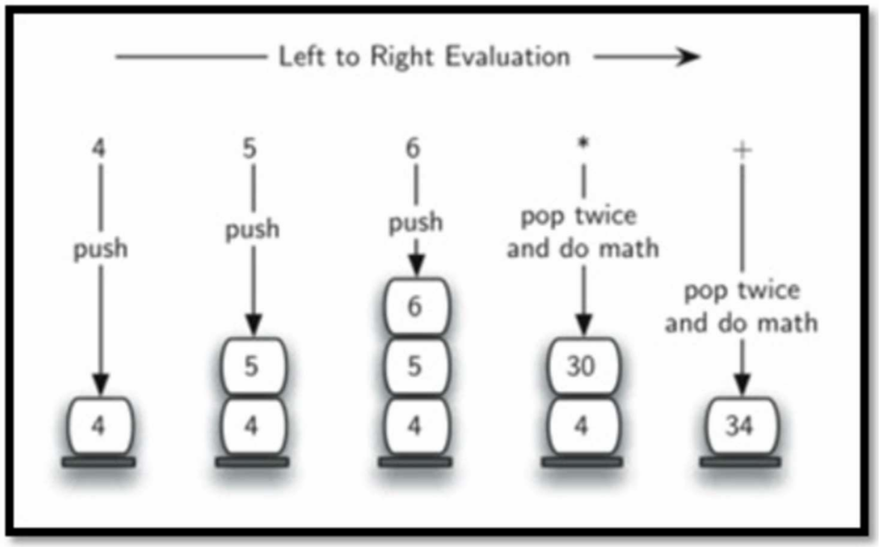
3) When the expression is ended, the number in the stack is the final answer.

Algorithm: To Evaluate Postfix expression.

```
Step 1: Add a ")" at the end of the
        postfix expression
Step 2: Scan every character of the
        postfix expression and repeat
        Steps 3 and 4 until ")"is encountered
Step 3: IF an operand is encountered,
        push it on the stack
        IF an operator O is encountered, then
        a. Pop the top two elements from the
           stack as A and B as A and B
        b. Evaluate B O A, where A is the
           topmost element and B
           is the element below A.
        c. Push the result of evaluation
           on the stack
        [END OF IF]
Step 4: SET RESULT equal to the topmost element
        of the stack
Step 5: EXIT
```

Example:

Postfix Expression: 456*+ = 34 Ans



Step	Input Symbol	Operation	Stack	Calculation
1.	4	Push	4	
2.	5	Push	4,5	
3.	6	Push	4,5,6	
4.	*	Pop(2 elements) & Evaluate	4	5*6=30
5.		Push result(30)	4,30	
6.	+	Pop(2 elements) & Evaluate	Empty	4+30=34
7.		Push result(34)	34	
8.		No-more elements(pop)	Empty	34(Result)

Result = 34

Program:

Input & Output:

Conclusion:

Sign and Remark:

R1	R2	R3	R4	Total Marks	Signature
(3)	(5)	(4)	(3)	(15)	