

MIT WORLD PEACE UNIVERSITY

Fundamental Data Structures
Second Year B. Tech, Semester 1

EXPRESSION EVALUATION USING STACK

PRACTICAL REPORT
ASSIGNMENT 8

Prepared By

Krishnaraj Thadesar
Cyber Security and Forensics
Batch A1, PA 20

November 28, 2022

Contents

| | |
|---|----------|
| 1 Aim | 1 |
| 2 Objectives | 1 |
| 3 Problem Statements | 1 |
| 4 Theory | 1 |
| 5 Platform | 1 |
| 6 Input | 1 |
| 7 Output | 1 |
| 8 Test Conditions | 1 |
| 9 Code | 1 |
| 9.1 Pseudo Code | 1 |
| 9.2 C Implementation of Problem Statement | 1 |
| 9.3 Input and Output | 3 |
| 10 Time Complexity | 4 |
| 11 Conclusion | 4 |
| 12 FAQs | 4 |

1 Aim

Writing a C Program to Evaluate a Postfix Expression using Stack.

2 Objectives

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

3 Problem Statements

Write a C Program to Evaluate a Postfix Expression using Stack.

4 Theory

5 Platform

Operating System: Arch Linux x86-64

IDEs or Text Editors Used: Visual Studio Code

Compilers : gcc on linux for C

6 Input

The Postfix Condition

7 Output

The Evaluation of the Postfix Expression entered.

8 Test Conditions

1. The Postfix Expression and Checking its Evaluation.

9 Code

9.1 Pseudo Code

9.2 C Implementation of Problem Statement

```
1 #include <stdio.h>
2
3 #define MAX_SIZE 30
4 int stack[MAX_SIZE];
5 int top = -1;
6
```

```
7 int powerr(int a, int power){
8     int i, result = 1;
9     for(i = 0; i < power; i++){
10         result *= a;
11     }
12     return result;
13 }
14
15 int isFull()
16 {
17     if (top == MAX_SIZE - 1)
18         return 1;
19     else
20         return 0;
21 }
22 int isEmpty()
23 {
24     if (top == -1)
25         return 1;
26     else
27         return 0;
28 }
29
30 int push(int item)
31 {
32     if (!isFull())
33     {
34         top++;
35         stack[top] = item;
36     }
37     else
38     {
39         printf("\nSTACK OVERFLOW!\n");
40     }
41 }
42 int pop()
43 {
44     if (isEmpty())
45     {
46         printf("Stack is Empty \n\n STACK UNDERFLOW!!");
47         return 0;
48     }
49     else
50     {
51         // printf("Removed this thing %c\n", stack[top]);
52         top--;
53         return stack[top + 1];
54     }
55 }
56
57
58 int calc(int op1, int op2, char op)
59 {
60     int ans;
61     switch (op)
62     {
63     case '+':
64         ans = op1 + op2;
65         break;
```

```
66     case '-':
67         ans = op1 - op2;
68         break;
69     case '*':
70         ans = op1 * op2;
71         break;
72     case '/':
73         ans = op1 / op2;
74         break;
75     case '%':
76         ans = op1 % op2;
77         break;
78     case '^':
79         ans = powerr(op1, op2);
80         break;
81     default:
82         printf("\nInvalid operator");
83         break;
84 }
85 return ans;
86 }
87
88 void eval(char post[MAX_SIZE])
89 {
90     int z = 0, ans = 0, op1, op2;
91     for (int i = 0; post[i] != 0; i++)
92     {
93         if(post[i] >= 97 && post[i] <= 122)
94         {
95             printf("\nEnter value of %c: ", post[i]);
96             scanf("%d", &z);
97             push(z);
98         }
99         else
100         {
101             op1 = pop();
102             op2 = pop();
103             ans = calc(op2, op1, post[i]);
104             push(ans);
105         }
106     }
107     printf("\nAnswer = %d\n", stack[top]);
108 }
109
110 int main()
111 {
112     char post[30];
113     printf("Enter the postfix expression: ");
114     scanf("%s", post);
115     eval(post);
116 }
```

Listing 1: Main.Cpp

9.3 Input and Output

```
1 Enter the postfix expression: abc+-cdb/++b^c+
2
3 Enter value of a: 6
```

```
4
5 Enter value of b: 2
6
7 Enter value of c: 3
8
9 Invalid operator
10 Enter value of c: 3
11
12 Enter value of d: 8
13
14 Enter value of b: 2
15
16 Enter value of b: 2
17
18 Enter value of c: 3
19
20 Answer = 52
```

Listing 2: Output

10 Time Complexity

- **Insertion:** $O(1)$
- **Deletion:** $O(1)$
- **Searching:** $O(n)$

11 Conclusion

Thus, implemented postfix expression evaluation using stack data structure.

12 FAQs

1. **How prefix expression evaluation works?**
2. **What is the advantage of prefix and postfix expressions?**