22-01-2026 - Apply stack operations to solve problems in various domains.

22 January 2026 14:03

=> a + (b * c)

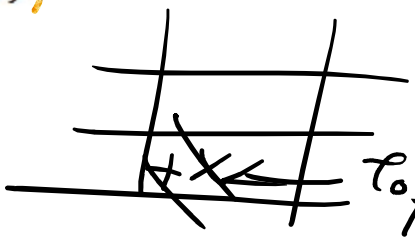
a + b c *

postfix : "abc*+"(\0) a b c * +

infix = a * b + c

postfix = a b * c +

```
while (infix[i] != '\0') {
    if (isOperand(infix[i]))
        postfix[j++] = infix[i++];
    else {
        while (st.top != -1 &&
            precendence(infix[i]) <= precendence(st.S[st.top])) {
            postfix[j++] = pop(&st);
        }
        push (&st, infix[i++]);
    }
}
while (st.top != -1)
    postfix[j++] = pop (&st);
postfix[j] = '\0';
return postfix;
}
```



a * b + c

a b * c +

a b * c +

=> Evaluation of the Postfix Operation:

3 * 5 + 6 / 2 - 4

2 3 1 4

$15 + 3 - 4$

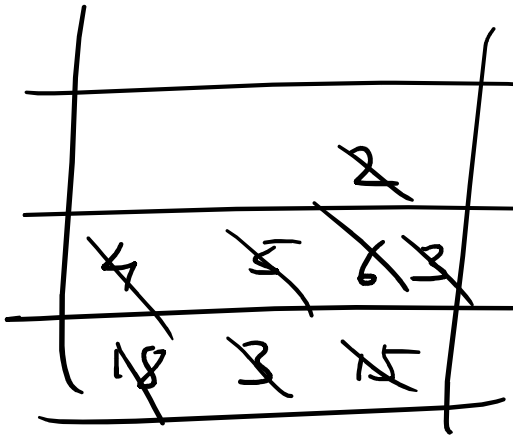
$18 - 4 = 14$

$\Rightarrow 3 \times 5 + 6/2 - 4$

$35\times + 62/ - 4$

$35\times62/+4-$

1) If operand push into stack.



$5 \times 3 = 15$

$6/2 = 3$

$15 + 3 = 18$

$18 - 4 = 14$

2) If operator pop-out the last 2 elements & perform the math. & then again push into the stack.

$\underline{4/2} + 8 \times 2 - 3$

$42/ + 8 \times 2 - 3$

$42/ + 82\times - 3$

4 2 / 8 2 x + 3 −

Symbol	Stack	Operation
4	4	
2	2, 4	
/	2	4 / 2 = 2
8	8, 2	
2	2, 8, 2	
x	16, 2	8 x 2 = 16
+	18	16 + 2 = 18
3	3, 18	
−	15	18 − 3 = 15