

Evaluation of Prefix

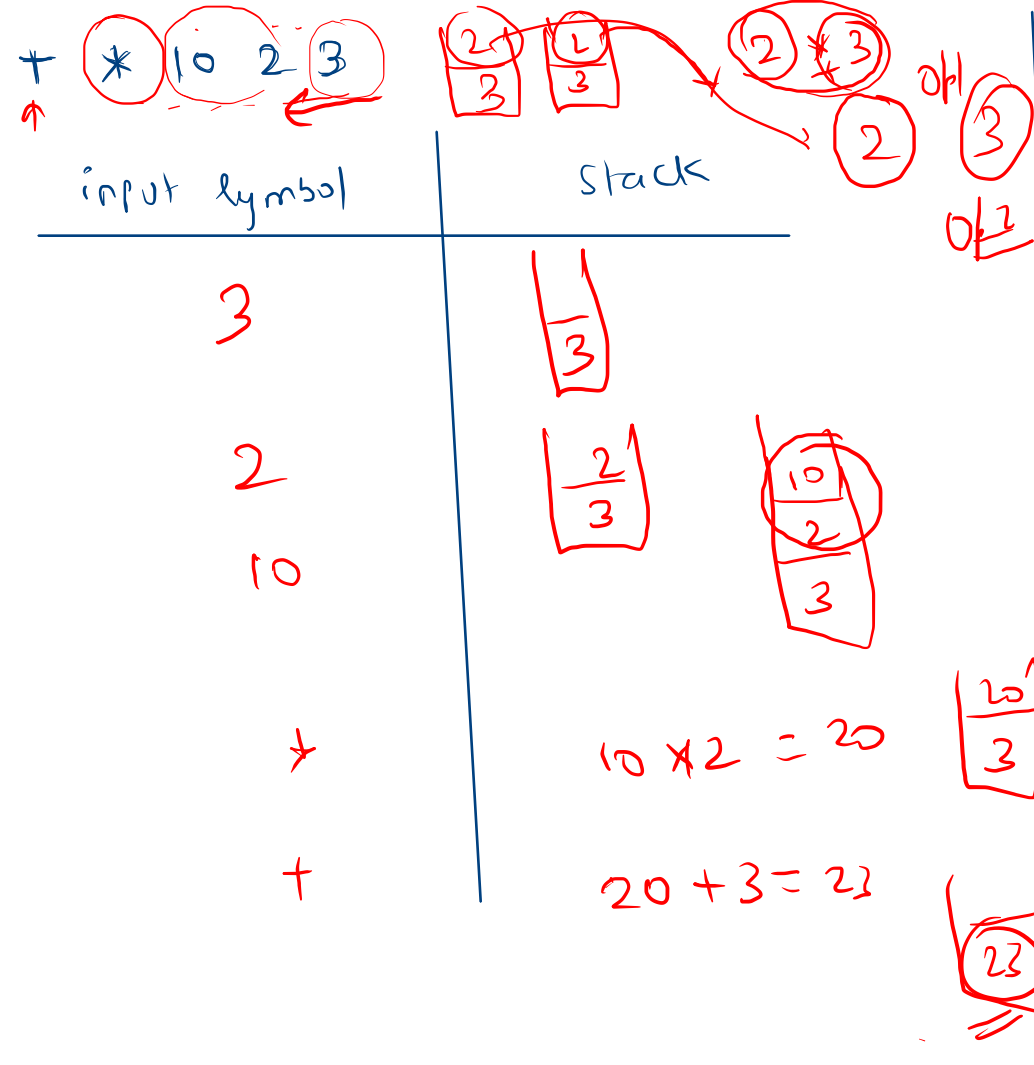
I/p: + * 10 2 3
o/p: 23

I/p: * + 10 2 3
o/p: 36

I/p: ^ 10 ^ 23
o/p: 10000 0000

Ex:

o/p:



- 1) create an Empty stack, st
- 2) Traverse through Every symbol x of given postfix [from Right to left]

a) if x is a operand, push to st

b) else (x is operator)

→ op1 = $st.top()$; $st.pop()$;
→ op2 = $st.top()$; $st.pop()$;
→ compute op1 x op2 and push the result to st .

3) Return $st.top()$;

Ex:

o/p:

⇒

* + 10 2 3

input symbol	stack
3	<div>3</div>
2	<div>2 3</div>
+ 10	<div>10 2 3</div> $10 + 2 = 12$
*	<div>12 3</div> $12 * 3 = 36$
	<div>36</div>

- create an Empty stack, **st**
- Traverse through Every symbol **x** of given postfix [from Right to left]

a) if **x** is a operand, push to **st**

b) **else** (**x** is operator)

→ $op1 = st.top(); st.pop();$

→ $op2 = st.top(); st.pop();$

→ compute $op1$ **x** $op2$ and push the result to **st**.

② Return $st.top();$