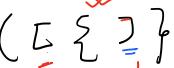
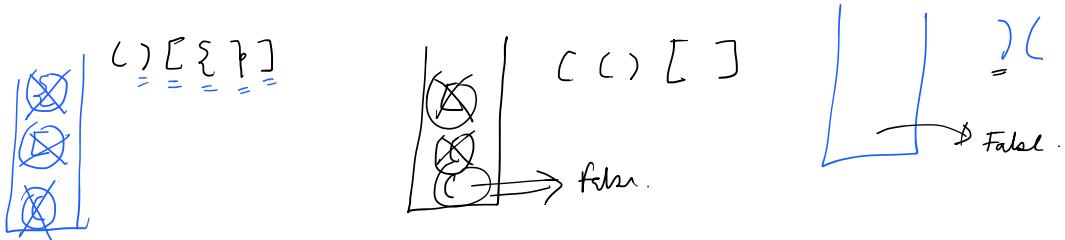


eg  \Rightarrow true 

eg  \Rightarrow false. eg  \Rightarrow true

eg  \Rightarrow false

if closing Brackets \Rightarrow check if the last opening bracket matches.



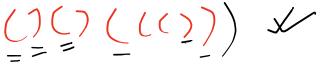
```

for i <= 0 .. n-1 {
    curchar = s.charAt(i)
    if isOpenB(curchar) {
        stack.push(curchar)
    } else {
        if stack.isEmpty()
            not false
        if curchar == topStack()
            pop()
        else
            not false
    }
}
if st.isEmpty()
    not True
else
    not False.
    
```

$Tc = \Theta(n)$
 $Sc = \underline{\Theta(n)}$

eg 

Q what will happen if we get only '}' type of Bracket.

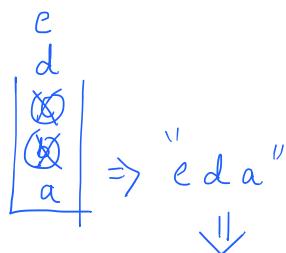
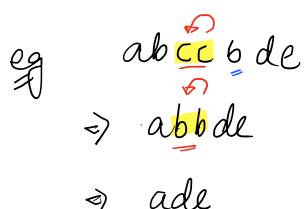
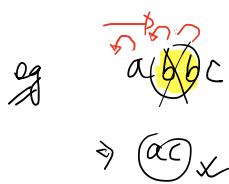
eg  \times

eg  $\Rightarrow \times$.

Given a char arr 's'

Remove equal pair of consecutive chars.

Multiple times till possible and of the final air



for $i \leftarrow 0 \dots n-1$ {

if stack.isEmpty()

stack.push (arr[i])

else if ($\text{stack_peek}() == \text{cur_char}$)

stack.pop();

else
stack.push();

$O(n)$

eq abab . . - ab

ansString = " ";

$$T_C < o(N)$$

while (! stack.isEmpty) {

char = stack.peek();

Stack.pop();

↳ amsting.appended(cmlines);

21

$$S_C = O(N)$$

$$SL = O(N) = \chi$$

Infix Expression

Post Exp.

operand1	op^r	operand2
----------	--------	----------

$$\Rightarrow 2 + 3$$

$$\Rightarrow 5 * 4$$

operand1	op	operand2
----------	----	----------

$$\Rightarrow 2 \ 3 \ +$$

$$\Rightarrow 5 \ 4 \ *$$

eg $2 * (3 - (6 / 2))$ Infix

$$\Rightarrow 2 \ 3 \ 6 \ 2 \ 1 \ - \ *$$

eg $a * (b - c)$

$$\Rightarrow a \ b \ c \ - \ *$$

Given a postfix expression Evaluate it.

eg $\underline{1} \ 0 \ \underline{5} \ = \ \Rightarrow 10 - 5 \Rightarrow 5$

$$\underline{2 \ 1} \ * \ 3 \ + \ \Rightarrow 2 * 1 \Rightarrow 2$$

$$\begin{array}{r} \underline{2 \ 3} \ + \\ - \end{array} \ \Rightarrow 2 + 3 \Rightarrow 5 \times$$

eg $\underline{\underline{3 \ 5}} \ \underline{\underline{+}} \ \underline{\underline{2}} \ - \ 2 \ 5 \ * \ -$

perform operatn on last 2 operands.

$$\begin{array}{r} \underline{\underline{8}} \ \underline{\underline{2}} \ - \\ - \end{array} \ 2 \ 5 \ * \ -$$

$$3 + 5 = 8 \quad 6 \quad \underline{\underline{2}} \ \underline{\underline{5}} \ \underline{\underline{*}} \ -$$

$$8 - 2 = 6$$

$$6 \quad \underline{\underline{10}} \ \underline{\underline{-}}$$

$$2 * 5 = 10$$

$$6 - 10 = -4$$

$$6 - 10 = -4$$

for $i \leftarrow 0 \dots n-1$ {

if $arr[i]$ is operand

push($arr[i]$)

else

$y = stack.pop()$

$x = arr[y]$

$x = \text{stack.pop()}$

$\text{res} = x \text{ op}^{\sigma} y$

$\text{push}(\text{res})$

}

st stack.peek();