

Page \_\_\_\_\_

# INFIX to POSTFIX CONVERSION

INFIX

$$(a \wedge b) / (a - b)$$

POSTFIX

$$xy \wedge 5zx / 2 +$$

PREFIX

$$+ ab$$

operator  $\rightarrow +, -, *, /, \wedge, \dots$

operand  $\rightarrow a, b, c, d, \dots$

Operator precedence

1) Parenthesis

{ } , ( ) , [ ]  
^

2) Exponents

3) Multiplication, Division : \* , /

4) Addition & Subtraction + , -

## Infix to Postfix

↳ Operator Precedence  
→ Associativity rules.

↓  
Associativity comes into picture when two operators are of same precedence.  
Eg:  $\times, /, +, -$

Eg

$$D = A + B * C.$$

Operators  $\Rightarrow =, +, *$   
Operand  $\Rightarrow D, A, B, C$

Step 1)  $D = A + B * C *$

2)  $D = A B C * +$

3)  $D A B C * + =$



## Infix to Postfix using stack

Date \_\_\_\_\_  
Page \_\_\_\_\_

- 1) Read all the symbols one by one from left to right in the given infix expression.
- 2) If the reading symbol is operand then directly print it to the result (op).
- 3) If the reading symbol is left parenthesis '(' then push it on to the stack.
- 4) If the reading symbol is right parenthesis ')' then pop all the contents of stack until respective left parenthesis is popped & print each popped symbol to the ~~the~~ result.
- 5) If the reading symbol is operator (+, -, \*, / etc) then push it on to the stack. However, first pop the operators which are already on the stack that have higher or equal precedence than current operator & print them to the result.

Eg:

$$(A+B) * (C-D)$$

Reading character

STACK

Postfix Expression

1)

Initially



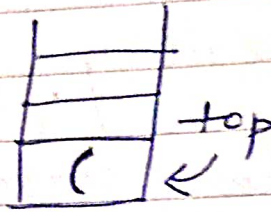
Stack is empty

Empty

2)

(

push 'C'



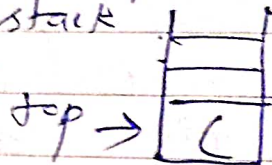
Empty

3)

A

A is an operand so don't push it in stack

A

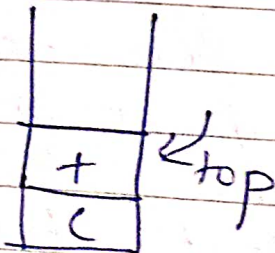


4)

+

'+' is operator. so push.

Push '+'

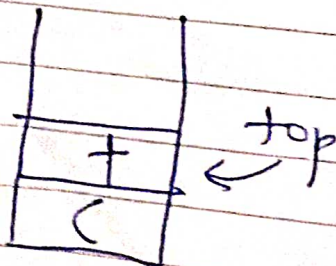


A

5)

B

B is an operand so put it in o/p



AB

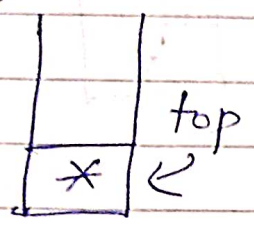


6) ) pop all the operators till closing paranthesis is open



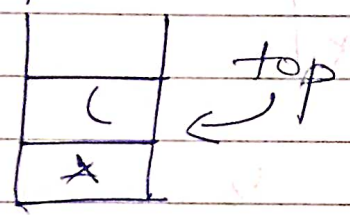
AB+

7) \* Push '\*' since operator



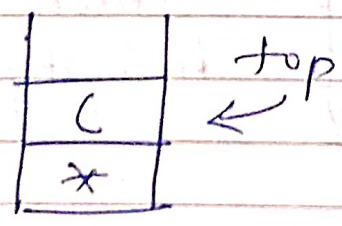
AB+

8) ( Push '(' as it is an operator



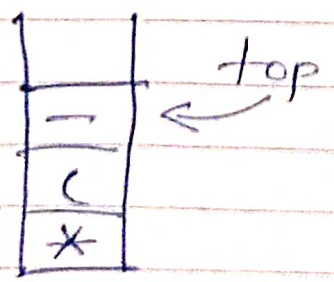
AB+

9) c c is an operand so put it in opp.



AB+c

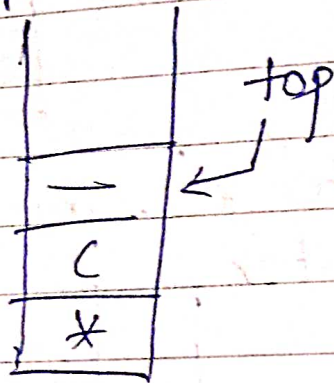
10) - Push '-'



AB+c

11) D

D is an operand  
so put it in o/p.

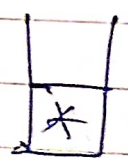


AB+CD

12)

)

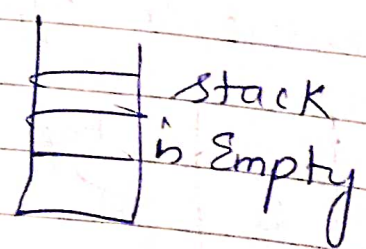
pop everything  
fill ^



AB+CD-

When there is no more operands/  
operators pop everything out

13) =



AB+CD-\*