

# Infix to postfix conversion

$A+B$  Infix

operands  $\rightarrow A, B, +, /, *, (, ) \dots$

$AB+$  postfix

operator  $\rightarrow +, -, /, *, (, ) \dots$

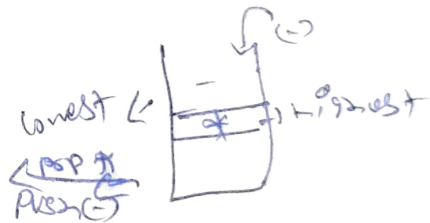
$+AB$  postfix

## Rules:

- Priority of operator:  $\therefore 1 \Rightarrow$  highest priority.  
 $*/ \Rightarrow$  next priority.  
 $+,- \Rightarrow$  lowest priority.

- No two operators of same priority can stay together in stack column.  
 $\rightarrow$  pop the operators  
 $\text{eg: } / *$

- Lowest priority cannot be placed before highest priority.



## Eg:

Postfix  $\rightarrow (A+B/C * (A+C) - F)$

Symbol                      Stack                      Postfix.

operand - postfix  
operator - stack.

(	(	
A	(	A
+	(+	A
B	(+)	AB
/	(+)	AB
c	(+)	ABC
*	(+)	ABC
C	(+)	ABC/
D	(+)	ABC/
+	(+)	ABC/D
	(+)	ABC/D



Symbol	Stack	Postfix
C	C + * / +	ABC / DC
)	( + * ( + ) POP.	ABC / DE
)	( + * ↓ Pop. High priority, so POP.	ABC / DE
-	( + - ↓ Same priority, so POP.	ABC / DE
-	( -	ABC / DE + * + F
F	( -	ABC / DE + * + F -



ABC / DE + \* + F -

### Evaluation of Postfix Expressions:

Suppose P - is an arithmetic expression written in postfix notation. ~~the~~ [ here of stack ]  
 $\text{Pop}(\text{AB} + \text{C}) / \text{E} / \text{F}$   
 here's operands.

Algorithm: Evaluate-Postfix (Stack, top, P)

- $\uparrow$  Stack; if is a linear array.
- $\uparrow$  top: top of the stack, where top = -1.
- P: the given postfix expression.

Step 1:  $\uparrow = 0$

Step 2: while (P[i] != '\0')      P[i] != '\0'

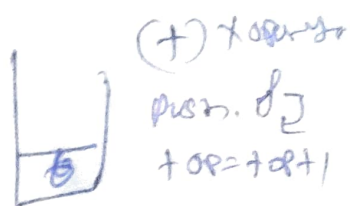
(a) if  $PCIF == \text{operand}$

$\rightarrow PCIF (6, 1, 9)$

{ push operand to the stack }

(i)  $top = top + 1$

(ii)  $stack[top] = PCIF$



(b)

if  $PCIF == \text{operator}$

$PCIF = +, *, -, \dots$

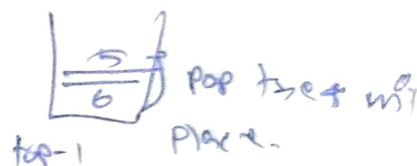
{ pop 2 top-most operands }

$P = 6 \ 5 \ 3 /$   
P0 P1 P2 P3 P4

(i)  $A = stack[top]$

(ii)  $top = top - 1$

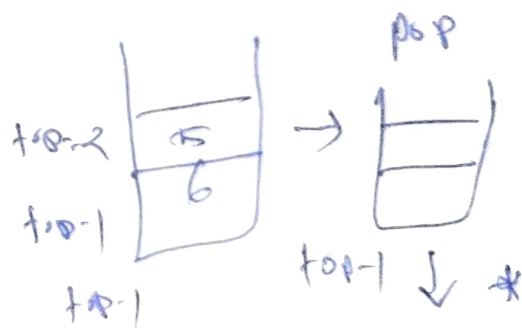
(iii)  $B = stack[P]$



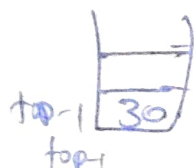
Evaluate  $(B * A)$  and put the result into the stack top.

(iv)  $stack[top] = B * A$

[End if]



(c)  $if (1)$   
{ End while }



Steps: [Display the result]

↓ print the  $stack[top]$

Step 4: Exit.

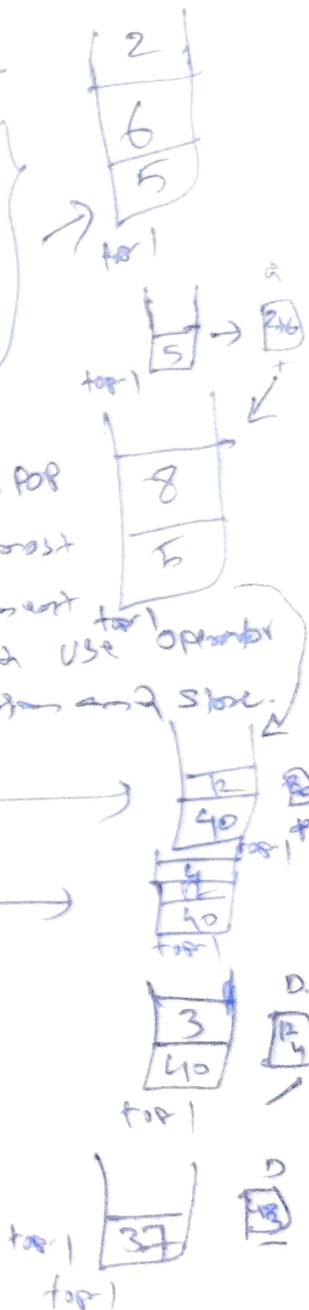
Example:

Evaluate the postfix notation.

P: 5, 6, 2, +, \*, 12, 4, /, -

Solution:

	Symbol scanned	Stack
(1)	5	5
(2)	6	5, 6
(3)	2	5, 6, 2
(4)	+	5, 8
(5)	*	40
(6)	12	40, 12
(7)	4	40, 12, 4
(8)	/	40, 3
(9)	-	37
(10)	)	37



Results

P = 37