Converting infix expression into postfix expression.

The steps involved to convert the infix expression into postfix expression will be -

step 1: Add the unique symbol '#' into stack and at the end of array infix

step 2: Scan the symbol of array infix one by one from left to right.

step 3: If symbol is left parenthesis '(' then add it to the stack.

step 4: If symbol is operand then add it to array postfix.

step 5: i) If symbol is operator then pop the operators which have same precedence or higher precedence than the operator which occurred.

ii) Add the popped operator to array postfix.

iii) Add the scanned symbol operator into stack.

step 6: i) If symbol is right parenthesis ')' then pop all the operators from stick until left parenthesis '(' in stack.

ii) Remove left parenthesis '(' from stack.

step 7: If symbol is '#' then pop all the symbols from stack and add them to array postfix except '#'

step 8: Do the same process until '#' comes in scanning array infix.

Eg. $A*(B+C\wedge D)-E\wedge F*(G/H)$

DATE

Initially # will be added in stack and at the end of infix expression.

so now the infix expression will be

$A*(B+C\wedge D)-E\wedge F*(G/H)\#$

step	Symbol	Operator in stack	Postfix expression
1	A	#	A
2	*	#*	A
3	(	#*(	A
4	B	#*(	AB
5	+	#*(+	AB
6	C	#*(+	ABC
7	^	#*(+^	ABC
8	D	#*(+^	ABCD
9	)	#*	ABCD^+
10	-	#-	ABCD^+*
11	E	#-	ABCD^+*E
12	^	#-^	ABCD^+*E
13	F	#-^	ABCD^+*EF
14	*	#-*	ABCD^+*EF^
15	(	#-*(	ABCD^+*EF^
16	G	#-*(	ABCD^+*EF^G
17	/	#-*(/	ABCD^+*EF^G
18	H	#-*(/	ABCD^+*EF^GH
19	)	#-*	ABCD^+*EF^GH/
20	#		ABCD^+*EF^GH/*-

So now the postfix expression is

$ABCD\wedge+*EF\wedge GH/*-$

Natural

Evaluation of Postfix expression :

The steps involved to evaluate postfix expression will be

Step 1: Add the unique symbol # at the end of array postfix.

Step 2: Scan the symbol of array postfix one by one from left to right.

Step 3: If symbol is operand then push it to stack.

Step 4: If symbol is operator then pop last two element of stack and evaluate it as - [top-1] operator [top]. and push it to stack.

Step 5: Do the same process until # comes in scanning

Step 6: Pop the element of stack which will be value of evaluation of postfix arithmetic expression.

eg. Let us take a postfix expression and evaluate it

① 4, 5, 4, 2, ^, +, *, 2, 2, ^, 9, 3, /, *, -

Initially '#' will be added at the end of postfix expression, so now the postfix expression will be

4, 5, 4, 2, ^, +, *, 2, 2, ^, 9, 3, /, *, -, #

DATE

step	Symbol	Operator in stack
1	4	4
2	5	4, 5
3	4	4, 5, 4
4	2	4, 5, 4, 2
5	^	4, 5, 16
6	+	4, 21
7	*	84
8	2	84, 2
9.	2	84, 2, 2
10	^	84, 4
11	9	84, 4, 9
12	3	84, 4, 9, 3
13	/	84, 4, 3
14	*	84, 12
15	-	72
16	#	

So, after evaluation of the postfix expression its value is 72.

Let us take the postfix expression in infix form and evaluate it

$4 * (5 + 4 \wedge 2) - 2 \wedge 2 * (9/3)$

$= 4 * (5+16) - 4 * 3$

$= 4 * 21 - 12$

$= 84 - 12$

$= 72.$

4, 5, 4, 2, ^, +, *, 2, 2, ^, 9, 3, /, *, -

= 4, 5, (4 ^ 2), +, *, (2 ^ 2), (9/3), *, -

= 4, (5 + 4 ^ 2), *, (2 ^ 2 * 9/3), -

= {4 * (5 + 4 ^ 2)}, {2 ^ 2 * 9/3}, -

= 4 * (5 + 4 ^ 2) - 2 ^ 2 * 9/3

H.W (2)

Convert the following postfix arithmetic expression into infix and evaluate it

45 + 36 * - 32 ^ + 82/3 * - 57 * +

Ans {4+5} {3+6} - {3^2} + {8/2} 3 * - {5*7} +

{(4+5) - (3*6)} (3^2) + {8/2 * 3} - {5*7} +

{(4+5) - (3*6) + (3^2)} {8/2*3} - {5*7} +

{(4+5) - (3*6) + (3^2)} - {8/2*3} {5*7} +

Infix → (4+5) - (3*6) + (3^2) - (8/2*3) + (5*7)

9 - 18 + 9 - 12 + 35

= 23

9.(5) Show the following postfix arithmetic expression evaluation in stack.

3 9 6 - ^ 6 2 / 5 * + 7 3 % -

Step	Symbol	Operator in Stack.
1	3	3
2	9	3 9
3	6	3 9 6
4	-	3 3
5	^	27
6	6	27 6
7	2	27 6 2
8	/	27 3
9	5	27 3 5
10	*	27 15
11	+	42
12	7	42 7
13	3	42 7 3
14	%	42 1
15	-	41
16	#	

Transform the following into infix expression.

① + * + − 7, 3, 2, 1, 3

② + A − * $ BCD / + EF * + KLI

③ AB − C + LPR − + $

Ans: ① + * + − 7, 3, 2, 1, 3

= + * + (7−3), 2, 1, 3

= + * {(7−3) + 2}, 1, 3

= + {(7 − 3) + 2} * 1, 3

Infix = (7 − 3 + 2) * 1 + 3

= (4 + 2) * 1 + 3

= 9

② + A − * $ BCD / + EF * + KLI

= + A − * (B $ C) D / (E + F) * (K + L) I

= + A − {(B $ C) * D} / (E + F) (K + L) * I

= + A − {(B $ C) * D} (E + F) / (K + L) * I

= + A (B $ C) * D − (E + F) / (K + L) * I

= A + (B $ C) * D − (E + F) / (K + L) * I

③ AB − C + LPR − + $

= {A − B} C + L{P − R} + $

= {(A − B) + C} {L + (P − R)} $

= A − B + C $ L + P − R