Stack

Stack Example

1) Polish Notation

There are three types of Notation

1) Infix Notation (A+B)
2) Prefix Notation (+AB)
3) Postfix Notation (AB+)

How to Convert Infix to Postfix

1) Solve brackets or Parenthesis
2) Solve Exponent or Power (^ or ↑)
3) Solve multiply or Divide [Left To Right].
4) Solve Addition or Subtract [Left to Right].

Convert this infix to Postfix

1st way (A+B) * C/D-E

Solution :-

(AB+) * C/D-E
T * C/D-E
(TC*)/D-E
M/D-E
(MD/)-E
K-E

T = AB+
M = TC*
K = MD/

KE-

=> MD|E-

=> TC*D|E-

=> AB+C*D|E-

II nd way ((A+B)*C|D-E)

step	Symbol Scanned	Stack	Expression
1	A	((	A
2	+	((+	A
3	B	((+	AB
4	)	(	AB+
5	*	(*	AB+
6	C	(*	AB+C
7	|	(|	AB+C*
8	D	(|	AB+C*D
9	-	(-	AB+C*D|
10	E	(-	AB+C*D|E
11	)		AB+C*D|E-

Sol we thos Infix to Postfix

12, 4, /, 5, 4, +, x, 9, /

input	Action	stack	Intermediate Result
12	Push	12	
4	Push	12, 4	
/	~~Push~~	~~3, 5, 4~~	
	Pop 2 operand	Empty	12/4 = 3
	Push Result	3	
5	push	3, 5	
4	push	3, 5, 4	
+	Pop 2 operand	3	5 + 4 = 9
	push Result	3, 9	
*	Pop 2 operand	Empty	9 * 3 = 27
	push Result	27	
9	push	27, 9	
/	Pop 2 operand	Empty	27/9 = 3
	push Result	(3)	

O/P => 3

Convert this infix Expression to Post fix.

(A + (B * C - (D / E ↑ F) × G) × H)

step	Symbol scanned	stack	Expression
1	A	(	A
2	+	(+	A
3	(	(+(	A
4	B	(+(	AB
5	*	(+(*	AB
6	C	(+(*	ABC
7	-	(+(-	ABC*
8	(	(+(-(	ABC*
9	D	(+(-(	ABC*D
10	/	(+(-(/	ABC*D
11	E	(+(-(/	ABC*DE
12	↑	(+(-(/↑	ABC*DE
13	F	(+(-(/↑	ABC*DEF
14	)	(+(-	ABC*DEF↑/
15	×	(+(-×	ABC*DEF↑/
16	G	(+(-×	ABC*DEF↑/G
17	)	(+	ABC*DEF↑/G*-
18	×	(+×	ABC*DEF↑/G*-
19	H	(+×	ABC*DEF↑/G*-H
20	)		ABC*DEF↑/G*-H*+

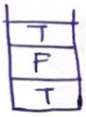
Evaluate the following Expression in Postfix form using a stack

True, false, true, not, false, true, or, not and, or, and

1) true is an operand
Push into Stack

2) false is an operand
push into stack

3) true "



4) not operator
Pop 1 operand
Not True => false

5) false "

6) True "

7) or is an operator
Not 2 operands
T or F => T

8 Not operator

9 and

10 or

11 and

Ans

Stack

10.4 $(A + ((B+C) + (D+E) * F)/G)$

step	Symbol Scanned	stack	Expression
1	A	(	A
2	+	(+	A
3	(	(+(	A
4	(	(+((	A
5	B	(+((	AB
6	+	(+((+	AB
7	C	(+((+	ABC
8	)	(+(	ABC+
9	+	(+(+	ABC+
10	(	(+(+(	ABC+
11	D	(+(+(	ABC+D
12	+	(+(+(+	ABC+D
13	E	(+(+(+	ABC+DE
14	)	(+(+	ABC+DE+
15	*	(+(+*	ABC+DE+
16	F	(+(+*	ABC+DE+F

17	)	(+	ABC+DE+F*+
18	/	(+/	ABC+DE+F*+
19	Gt	(+/	ABC+DE+F*+Gt
20	)		ABC+DE+F*+Gt/+

[4, 10, 5, +, *, 15, 3, /, -]

input	Action	stack	
4	Push	4	
10	Push	4, 10	
5	Push	4, 10, 5	
+	Pop 2 operand	4, 15	10+5 = 15
*	Pop 2 operand	60	4*2 = 8
15	Push	60, 15	
3	Push	60, 15, 3	
/	Pop 2 operand	60, 5	15/3 = 5
-	Pop 2 operand	55	8-5 = 3
		55	

15 15 3 2 + / 7 + 2 *

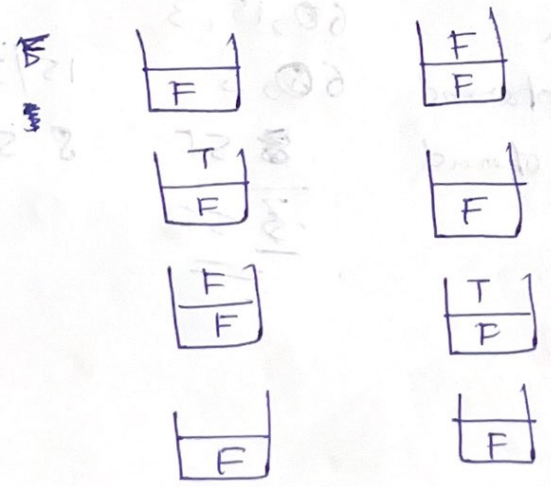
15	15
3	15, 3
2	15, 3, 2
+	15, 5
/	3
7	3, 7
+	10
2	10, 2
*	20

3+2 = 5

15/5 = 3

3+7 = 10

F, T, Not , AND, F, OR, T, AND



F

) (* (-

* (* (* - *

G

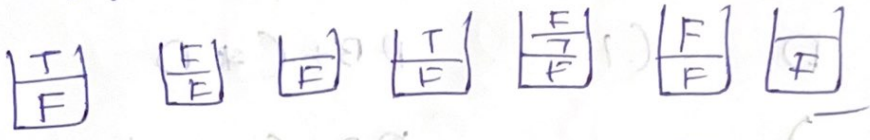
)

	ABC* DFF
	ABC* DEF 1/
	ABC* DEF 1/ G
	ABC* DEF 1/ G* - H
	* +

50, 60, +, 20, 10, −, *

50	50
60	50, 60
+	110
20	110, 20
10	110, 10
−	1100
*	

F, T, NOT, OR, T, F, AND, OR



(A+(B * C - (D/E ↑ F) * G) * H)

(	(	
A	(	A
+	(+	A
B	(+	AB
*	(*	AB+
C	(*	AB+
(	(+(	A
B	(+(	AB
*	(+(*	AB
C	(*(*	ABC
−	(*(-	ABC*
(	(*(-(	ABC*
D	(*(-(	ABC*D
/	(*(-(/	ABC*D
E	(*(-(/	ABC*DE
↑	(*(-(/↑	ABC*DE

Q ((A+B) * C/D)

	((	A
+	((+	A
B	((+	A B
)	(	A B +
*	(*	A B +
C	(*	A B + C
/	(/	A B + C *
D	(/	A B + C * D
)		A B + C * D /

Q ((A+B) * C/D + (E / F))/G

	((	A
+	((+	A
B	((+	A B
)	(	A B +
C	(	A B + C
/	(/	A B + C
D	(/	A B + C D
+	(+	A B + C D +
*	(*	A B +
C	(*	A B + C
/	(/	A B + C *
D	(/	A B + C * D
+	(+	A B + C * D /

E

↑

F

C+

C* ↑

A B + C * D / E

A B + C * D / E

A B + C * D E F ↑ * +