

Evaluations → ① Infix . ② Prefix ③ Postfix

operand → variable name

operator → +, -, *, /, ^ - - - etc

Humans are solve infix Expression, ex

B → Bracket }
 D → Divide }
 M → Multiplication }
 A → Addition }
 S → Subtraction }

Infix { $(5 + 4) * 2 = 9 * 2 = 18$
 $5 + 4 * 2 = 5 + 8 = 13$
 value1 operator value2

post fix

→

Left	Right	Node
<u>value1</u>	<u>value2</u>	<u>operator</u>
Ex → 5	4 2	+ *

prefix →

Node	Left	Right
<u>operator</u>	<u>val1</u>	<u>val2</u>
Ex' + *	5 4	2

Expression



(i) Evaluate

(ii) Conversion

Set - I

1.1 Infix

Evaluation

Evaluate

1.2 Infix

to prefix

1.3 Infix

to postfix

Set - II

2.1

Postfix Evaluation

2.2

Postfix to prefix

2.3

Postfix to infix

Evaluate

3.1

Prefix Evaluation

3.2

Prefix to postfix

3.3

prefix to infix

Set - III

Conversion

Infix Evaluation,

Expression →

$$\overset{\vee}{2} + \overset{\vee}{3} * 2 \ominus (3 + 4 * 1) = 1$$

↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑

$$5 - 2$$

$$((2 + 3 + 4) - 1 * 7)$$

14 ↑ ↑ 7 ↑

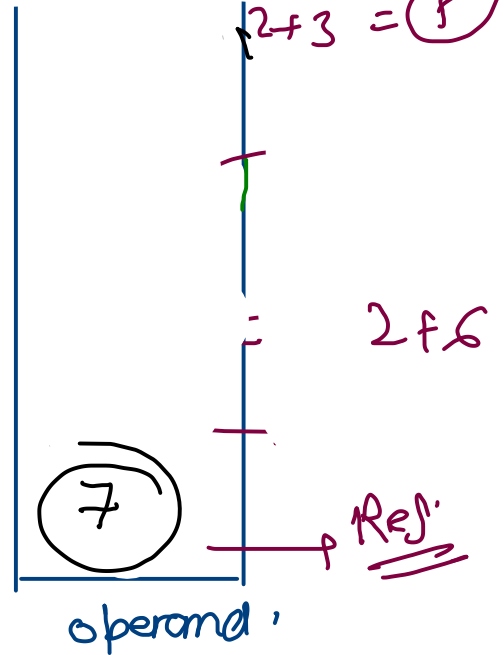
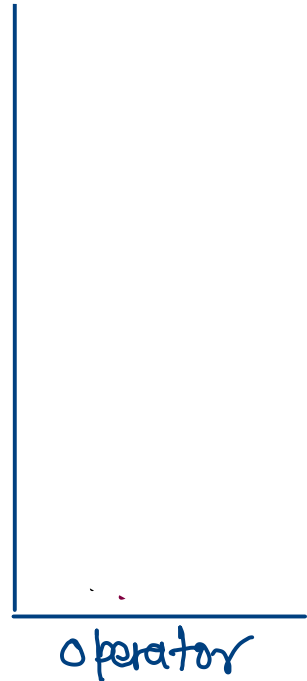
$$(2 + \cancel{3} + \cancel{4} + \cancel{1} + \cancel{8})$$

$$- 24 / 8$$

$$2 + 3 = 5$$

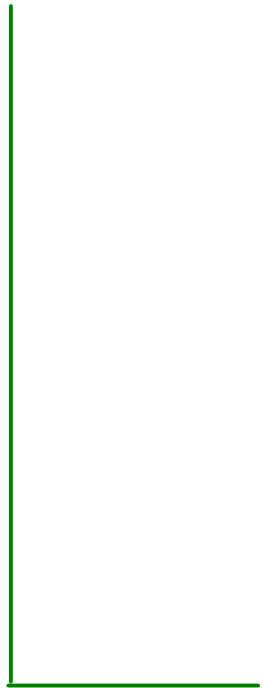
B	→	3
/	→	2
*	→	2
+	→	1
-	→	1

Greater priority operator
or lower priority operator
can pushed.



Infix Prefix

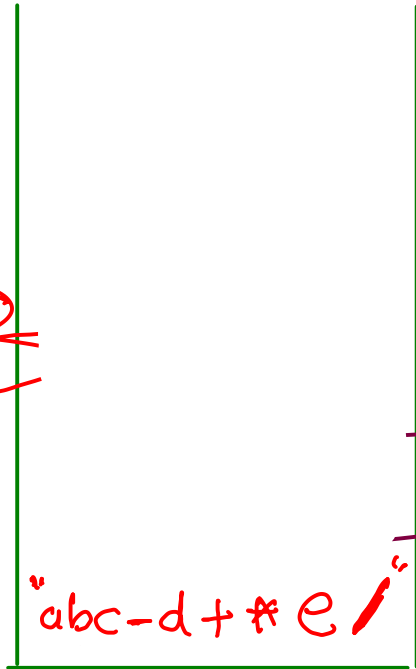
Postfix



Stack

↑↑↑↑↑↑↑

~~2~~



v.stack

op ~ '*'

val1 = 'a'

val2 = '+bcd'

op val1 val2

post } → val1 val2 op

/ * a + - b c d e

Post fix evaluation: →

val1 val2 operator

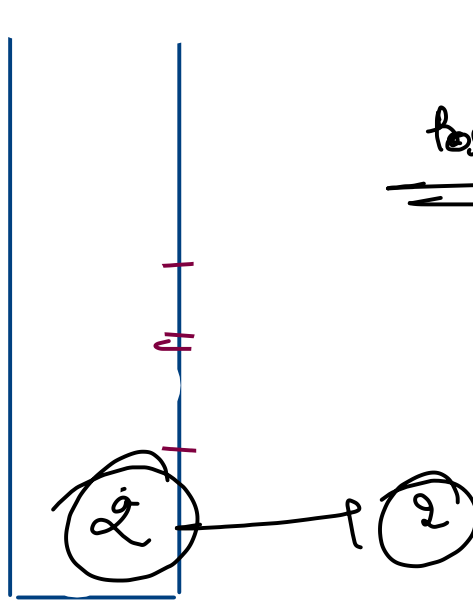
2 6 4 * 8 / + 3 -

↑ ↑ ↑ ↑ ↑ ↑ ↑

infix → ~~2~~ ~~6~~ * ~~8~~ / 8 - 3

post fix to infix

o/p val1 val2

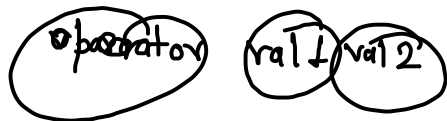


3.

$(2 + ((6 + 8) / 8))$

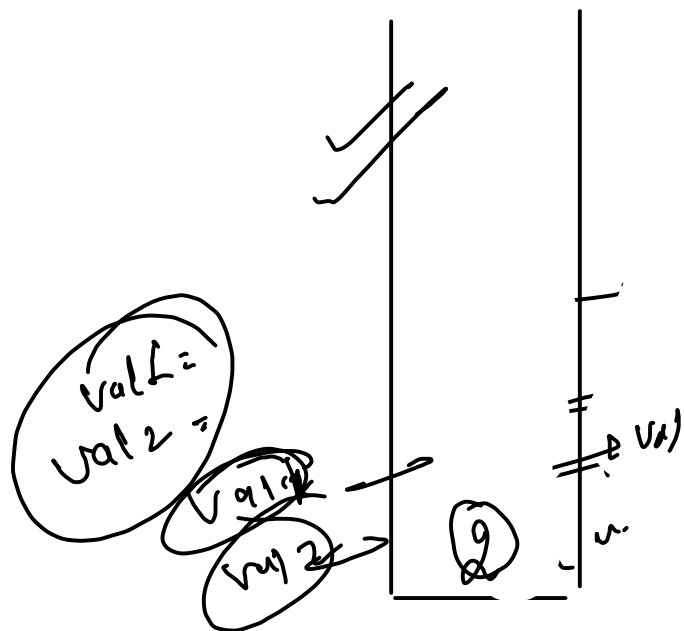
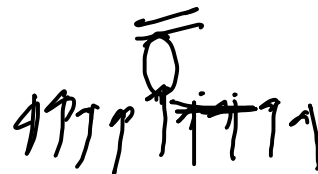
✓ → $(2 + ((6 + 8) / 8)) = 3$

Prefix Evaluation. →



val1 val2 op

val1 val2 op



Soln

$$\left(\left(2 + \left((6 * 4) / 8 \right) \right) - 3 \right) \quad 264 * 8 / + 3 -$$

In

Postfix