Evaluations and Conversions :→

4+3*2 → infix

equation

⇒ {{ (4+3) * 2 } → 7*2 = 14

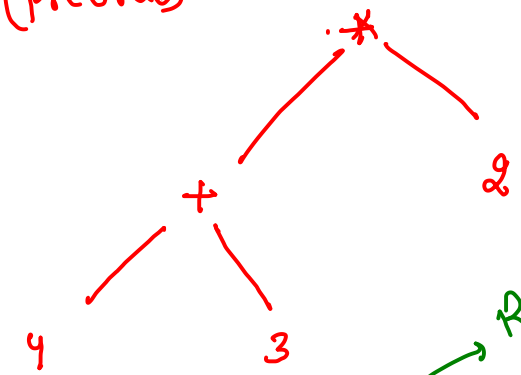
⇒ {{ 4 + (3*2) → 4+6 = 10

infix

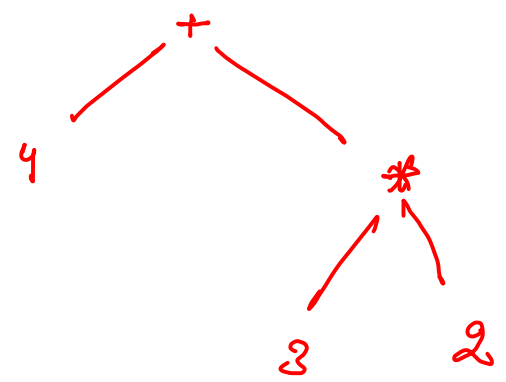
infix → (inOrder)

postfix → (postOrder)

prefix → (preOrde)



Result after solving



inOrder →	4 + 3 * 2	→ infix
preOrder →	* + 4 3 2	→ prefix
postOrder →	4 3 + 2 *	→ postfix

(2+3)

① Infix ⟶

1.1 Infix Evaluation

1.2 Infix to prefix

1.3 Infix to postfix ⟶ val1 operator val2

equation
- ① operators -
- ② operand

② Prefix ⟶

2.1 Prefix Evaluation

2.2 Prefix postfix ⟶ operator val1 val2

2.3 Prefix Infix

② Postfix ⟶

3.1 Postfix Evaluation ⟶ val1 val2 operator

3.2 Postfix to prefix

3.3 Postfix to Infix

Equation: $4+3*2$

BDMAS (Helping in Decision of priority)

B → Bracket

D → Division

M → Multiplication

A → Addition

S → Subtraction

Decreasing order of priority.

Bracket — P1

Division, Multiplication — P2

Addition, subtraction — P3

$P1 > P2 > P3$

How to decide for priority of same solution?

→ Travel from left to right

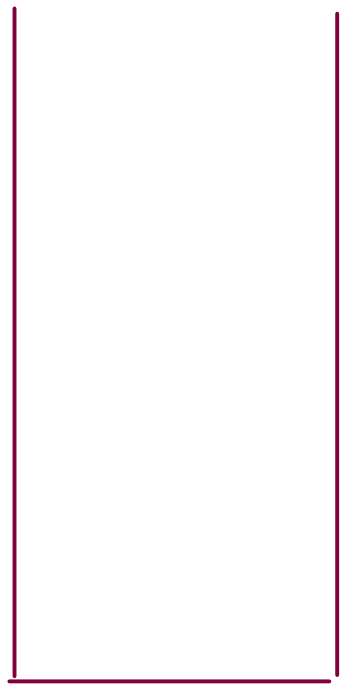
same priority, left → right

Eg → "2 + 3 * 2 - (3 + 4 * 1)"

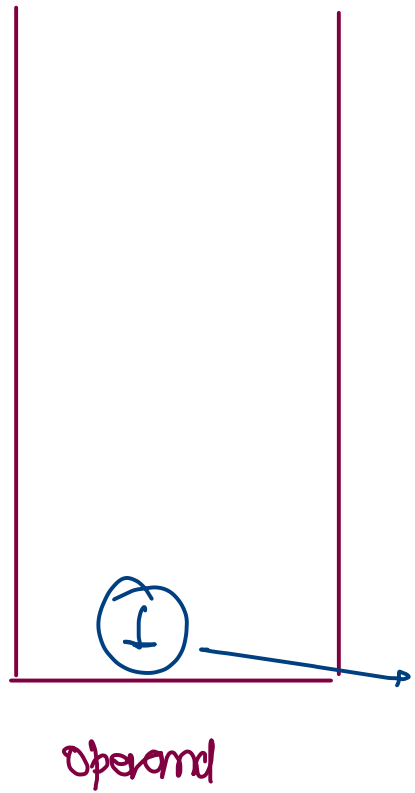
* , / → 2
+ , - → 1
Rest → 0

Infix → val1 op val2

operator = * + -
val1 = 4 3 8
val2 = 1 4 7
res = 4 7 1



operator



operand

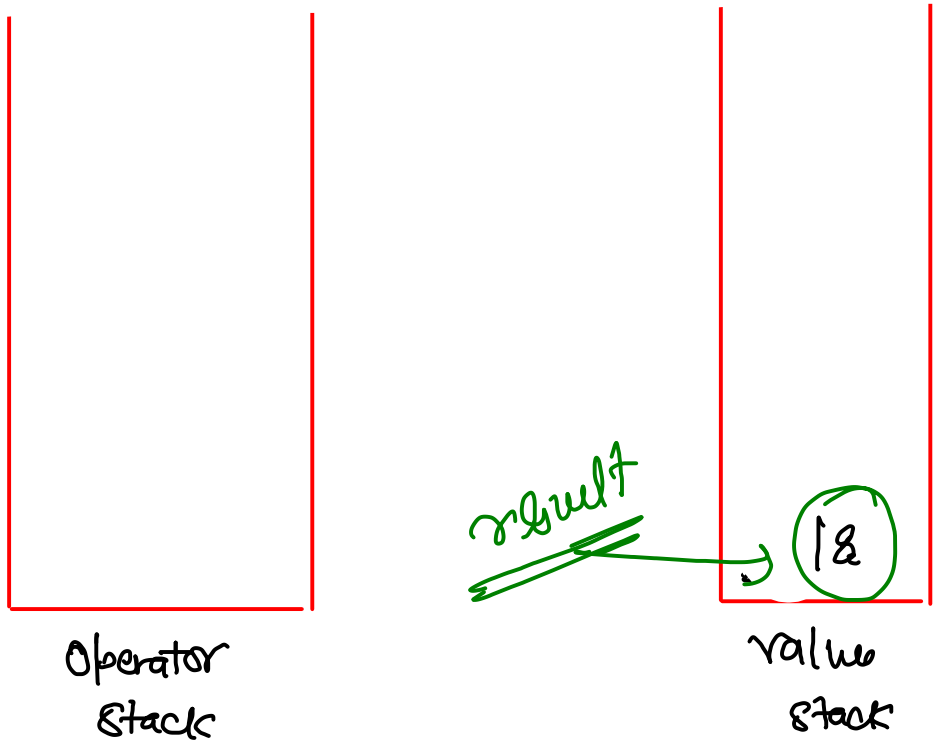
operand.pop()

higher priority operator can be pushed on lower priority op

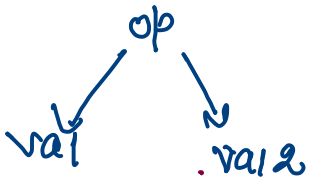
res = 18

Infix → "(7 - (1 + 2 * 3) + 6/2) + 9 * 2"

Ch = space continue
Ch = digit
Ch = opening bracket
Ch = closing bracket
Ch = operator.



operator = ~~*~~ +
val1 = ~~2~~ 0
val2 = ~~2~~ 18
rs = val1 ~~op~~ val2 = ~~18~~ 18



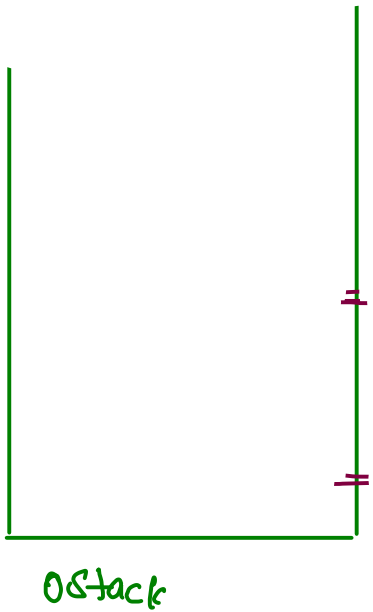
infix postfix → val1 val2 operator

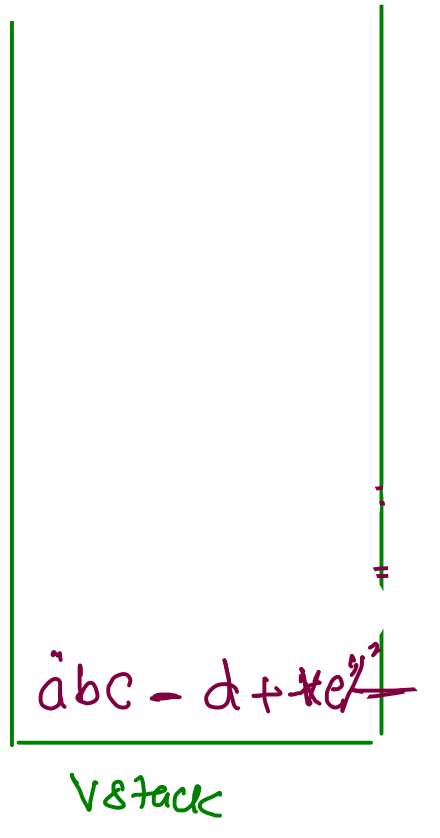
postfix → "abc-d+*e"

infix → val1 op val2

prefix → op val1 val2

postfix → val1 val2 op.





→ postfix

val1 val2 op.

"b c -"

"bc-d+"

"abc-d+*"

abc-d+* e

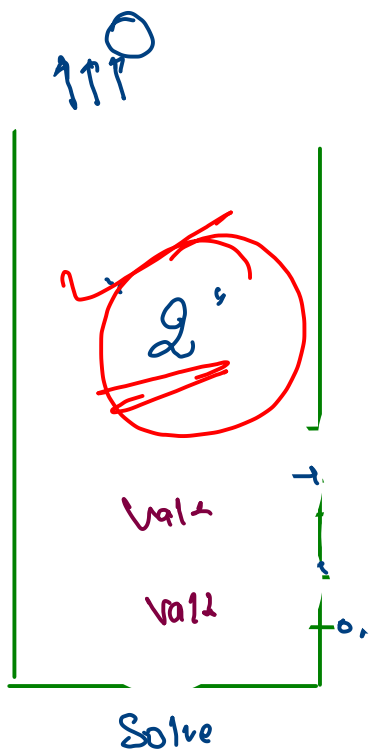
a * (b - c + d) / e → infix

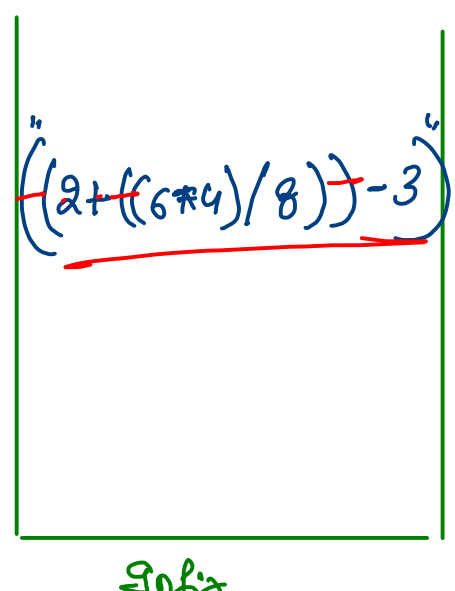
prefix → / * a + - b c d e op val1 val2

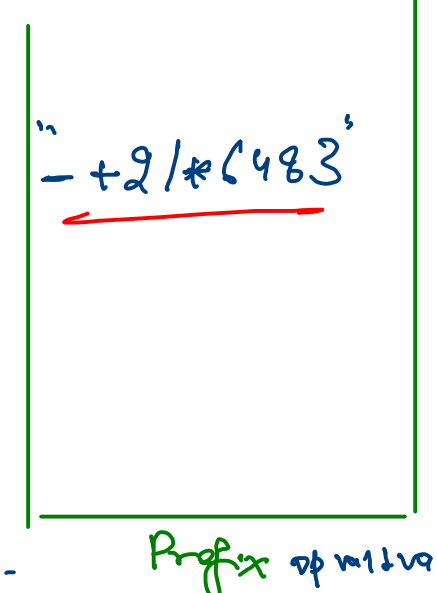
postfix → a b c - d + * e / val1 val2 op

S [ ]

these operators are arranged by their priority.

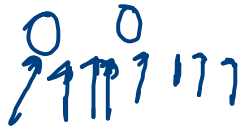


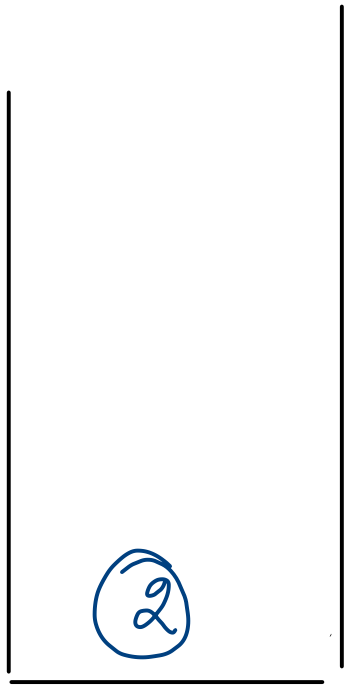




prefix ⟶ operator val val2.

/ * a + - b c d e





op = ~~*~~ ~~×~~ ~~÷~~ −

val1 = ~~6~~ ~~4~~ ~~2~~ 5

val2 = ~~4~~ ~~8~~ ~~3~~ 3

~~postfix~~ ⟶ val1 val2 op.

~~infix~~ ⟶ (val1 + op + val2)