Expression

- statement of a program which always give a result
- statement is combination of operators and operand
    Operands - values/variables
    Operators - mathematical symbols (+, -, *, /, %, power)
- eg a + b, a - b * c

Types of expressions
    1. Infix   :   a + b   (human)
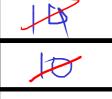   2. Postfix   :   a b +   (computer)
    3. Prefix   :   + a b   (computer)

()
power $
*, /, %
+, -

1. Postfix evaluation
2. Prefix evaluation
3. Infix to postfix conversion
4. Infix to prefix conversion

Postfix Evaluation

Postfix : 4 5 6 * 3 / + 9 + 7 -

left ⟶ right

Stack

③ 23 − 7 = 16

Result = 16

④ 14 + 9 = 23

③ 4 + 10 = 14

② 30 / 3 = 10

① 5 * 6 = 30

ASCII values

'0' - '9' => 48 - 57

'0' → 48   48 - 48 = 0

'1' → 49   49 - 48 = 1

'2' → 50   50 - 48 = 2

⋮   ⋮

Stack (column, top to bottom):
16, 7, 23, 9, 14, 10, 8, 30, 6, 5, 4

Prefix Evaluation

Prefix :- + + 4 / * 5 6 3 9 7

left ← right

result = 16

⑤ 23 − 7 = 16

④ 14 + 9 = 23

③ 4 + 10 = 14

② 30 / 3 = 10

① 5 * 6 = 30

16
23
14
4
10
30
5
6
3
9
7

Infix to Postfix conversion

Infix : 1 $ 9 + 3 * 4 - (6 + 8 / 2) + 7

left ———————————→ right

Postfix: 19$34*+682/+-7+

Infix to Prefix conversion

Infix : 1 $ 9 + 3 * 4 - (6 + 8 / 2) + 7

left ⟵ right

Expression : 7 2 8 16 + 43 * 91 $ + − +

Prefix : + − + $ 19 * 34 + 6 / 82) (

Infix to postfix and prefix conversion

Postfix

Infix : 1 $ 9 + 3 * 4 - (6 + 8 / 2) + 7

1 $ 9 + 3 * 4 - (6 + 8 2 /) + 7

1 $ 9 + 3 * 4 - 6 8 2 / + + 7

1 9 $ + 3 4 * - 6 8 2 / + + 7

1 9 $ 3 4 * + - 6 8 2 / + + 7

1 9 $ 3 4 * + 6 8 2 / + - + 7

1 9 $ 3 4 * + 6 8 2 / + - 7 +

Prefix

Infix : 1 $ 9 + 3 * 4 - (6 + 8 / 2) + 7

$ 1 9 + * 3 4 - (6 + / 8 2) + 7

$ 1 9 + * 3 4 - + 6 / 8 2 + 7

+ $ 1 9 * 3 4 - + 6 / 8 2 + 7

- + $ 1 9 * 3 4 + 6 / 8 2 + 7

+ - + $ 1 9 * 3 4 + 6 / 8 2 7