Stack

- stack is a linear data structure in which data is stored sequentially.
- in stack, data insertion and deletion is allowed from only one end (top).
- stack follows the principle of "Last In First Out"
- all operations of stack are done in O(1) time
- stack can be implemented using array or linked list

size = 5

Operations:

1. Add/Insert/Push

a. reposition top (inc)
b. add data at top index

2. Delete/Remove/Pop

a. reposition top (dec)

3. Peek (Collect)

a. read data from top index

Conditions:

Empty	Full
top == -1	top == size - 1

Stack Application

Expression Evaluation and Conversion

1. Postfix Evaluation
2. Prefix Evaluation
3. Infix to Postfix Conversion
4. Infix to Prefix Conversion

Expression:

- combination of operators and operands
- operators are mathematical operations like +, -, *, /, % or power
- operands are values/numbers/variables

e.g. a + b - c, a + b * c, a

Types

1. Infix	a + b	- human
2. Prefix	+ a b	- computer
3. Postfix	a b +	- computer

Priorities:

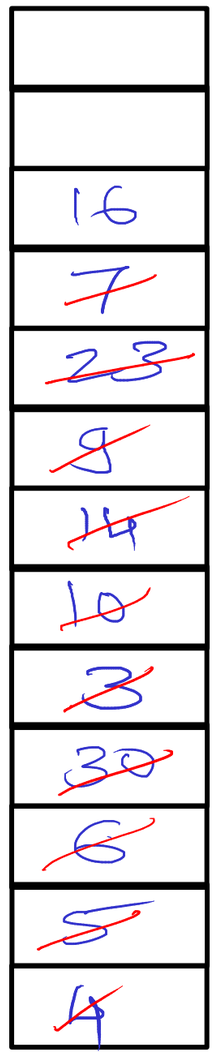
Postfix Evaluation

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

left ⟶ **right**

Result = 16

Stack



(5) 23 − 7 = 16

(4) 14 + 9 = 23

(3) 4 + 10 = 14

(2) 30 / 3 = 10

(1) 5 * 6 = 30

else

'0' - ascii 48 ⇒ 48 − 48 = 0

'1' - ascii 49 ⇒ 49 − 48 = 1

'2' - ascii 50 ⇒ 50 − 48 = 2

'3' - ascii 51 ⇒ 51 − 48 = 3

'4' - 52 ⇒ 52 − 48 = 4

.

.

.

Prefix Evaluation

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

left ⟵ **right**

Result = 16

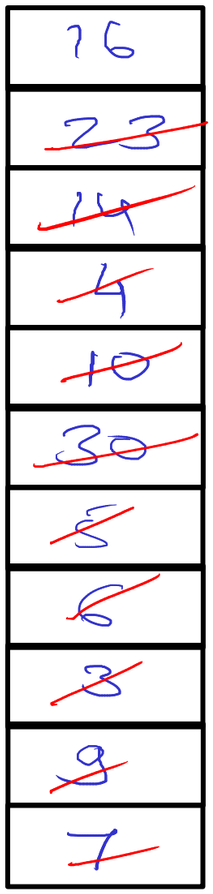
23 − 7
= 16

14 + 9
= 23

4 + 10
= 14

30 / 3
= 10

5 * 6
= 30

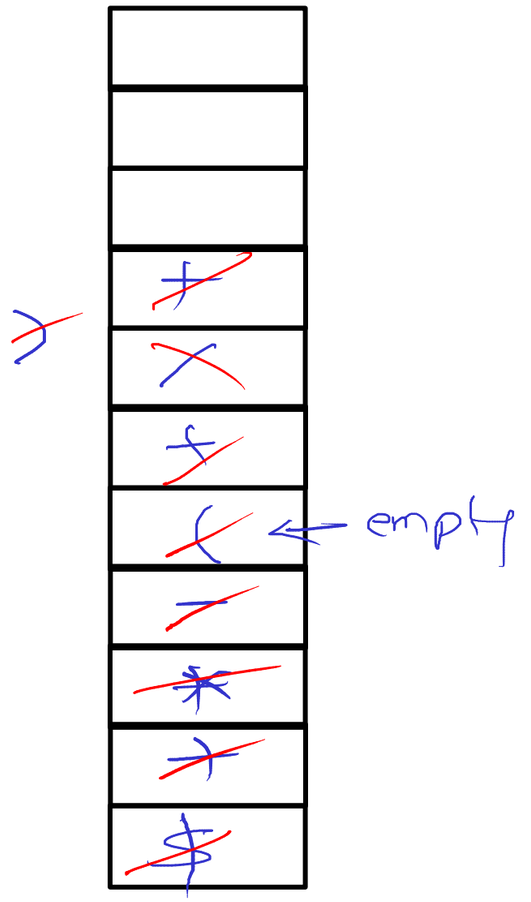


Infix to Postfix conversion

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

left ⟶ **right**

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



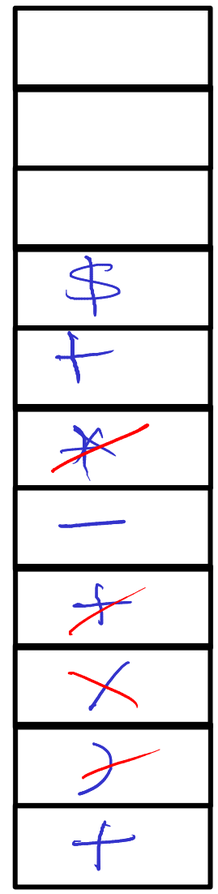
Infix to Prefix conversion

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

left ← **right**

Expression : 7 2 8 / 6 + 4 3 * 9 1 $ + − +

Prefix : + − + $ 1 9 * 3 4 + 6 / 8 2 7



Infix to postfix and prefix conversion

Postfix

③ ⑤ ④ ⑥ ② ① ⑦

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

1$9+3*4-(6+82/)+7

1$9+3*4-682/++7

19$+3*4-682/++7

19$+54*-682/++7

19$54*+-682/++7

19$34*+682/+-+7

19$34*+682/+-7+

Prefix

③ ⑤ ④ ⑥ ② ① ⑦

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

$19+*34-(6+/82)+7

$19+*34-+6/82+7

+$19*34-+6/82+7

-+$19*34+6/82+7

+-+$19*34+6/827