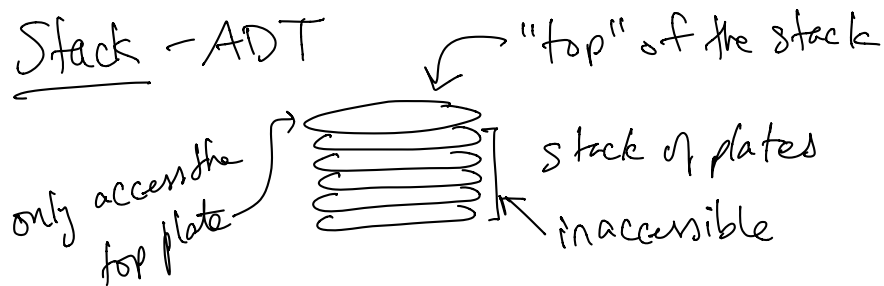


Finish up linked lists ✓

Stacks/Queues - ADTs

↳ restricted versions of lists. → only allow insertion/deletion at one/both ends of the structure.

Stack - ADT



LIFO =
last in, first
out

Two main operations

PUSH — add a new item to the top of the stack.

POP — removes (usually returns) the top item from the stack.

Other things: peek — check what the item is on the top of the stack, but don't remove it.

get size

check if the stack is empty.

ALL OPERATIONS THAT MODIFY A STACK ONLY HAPPEN AT ONE END (TOP).

Start w/ an empty stack.

STACK

Push 1

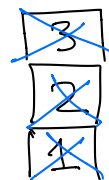
Push 2

Push 3

Pop → returns 3

Pop → return 2

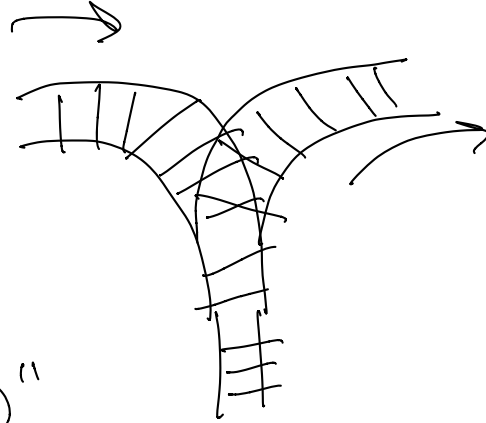
Pop → returns 1



Push 1
 Push 2
 Push 3
 Pop
 Push 4
 Pop
 Pop
 Push 5
 Pop
 Pop

Order of items that are removed from the stack

3, 4, 2, 5, 1

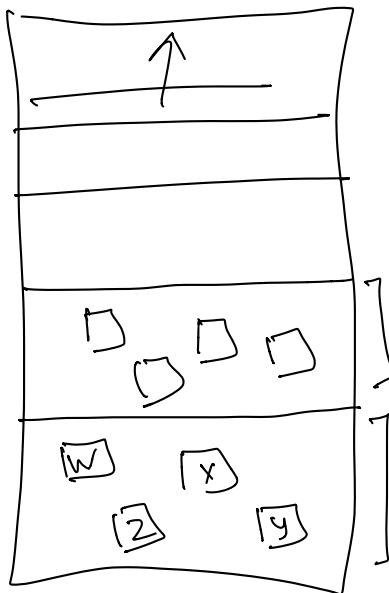


$$(x + (y - 3)) - (z + 4)$$

Red arrows indicate stack operations: two 'Push' arrows point to the 'y' and '3' in the inner expression, and a 'Pop' arrow points to the '3'.

$$(x + 1) \quad) x + 1$$

"The Stack"



f()

all do your variables in main()

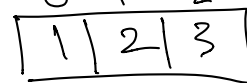
Implement a stack

Either w/ an array or linked list

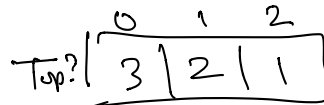
Array — where should the "top" of the stack be in the array?

Push — append

makes more sense



Top?

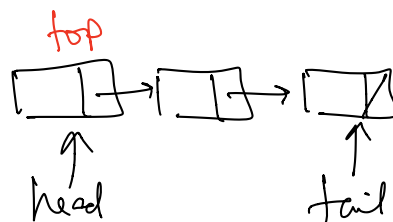
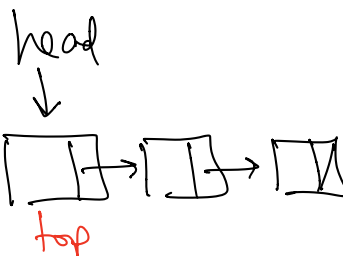


Pop — remove from RHS.

Linked List

Singly LL? → head? tail?

Doubly LL?



Postfix notation (Reverse polish notation)

3 4 +
7 2 *

} Parentheses are not needed

Infix notation

3 + 4
7 * 2

(1 + 2) * 3

1 + (2 * 3)

↓
1 2 + 3 *
→ → → →

↓
1 2 3 * +
→ → → →

Stack

* 3
2 3
1 3 (9)

Stack

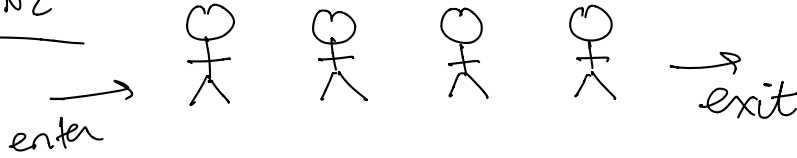
* 3
2 6
1 7

3
3
2
1

Queue

- all items are added at one end & removed from the other end.

WAITING LINE

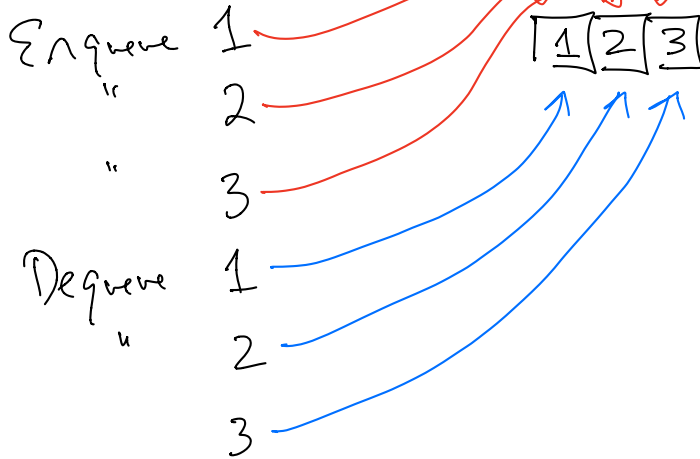


• Printer queue

Two main operations

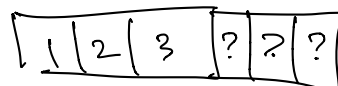
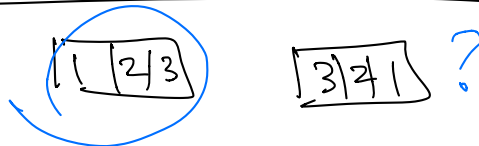
Enqueue: add an item to one end of the queue

Dequeue: remove an item from the other end of the queue



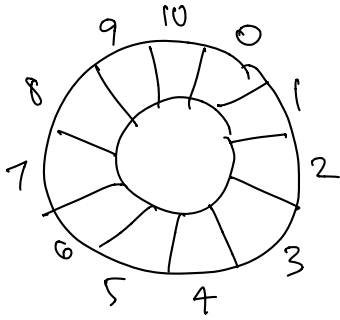
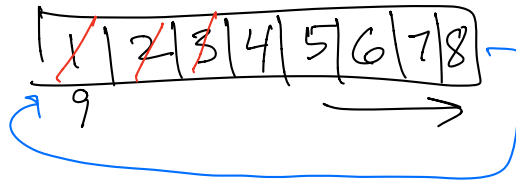
FIFO
first-in,
first-out

Array-based-queue



eng 1
eng 2
deg
eng 3
deg
eng 4

dog
eng 5 →



linked-lists-queue

SLL?

head?

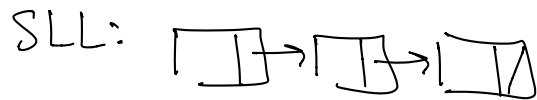
DLL?

head+tail?



add

delete



delete

add