

1.4 DATA TYPES & STRUCTURES · 1.4.2

Stacks & queues

LIFO vs FIFO, the pointers, and circular queues. Spec 1.4.2(b)(c).

01 Stack — LIFO

Rule Last In, First Out. Add/remove at the **top**.

Pointer `top` (−1 when empty).

push `top++` then `stack[top]=item`

pop read `stack[top]` then `top--`

peek return top item, don't remove.

Errors **overflow** (push when full), **underflow** (pop when empty).

02 Queue — FIFO

Rule First In, First Out.

Pointers `front`, `rear`, `size`.

enqueue add at the **rear**.

dequeue remove from the **front**.

Checks `isEmpty` (size=0), `isFull` (size=maxSize).

03 Linear vs circular queue

● Linear

Front space freed by dequeue is **wasted**; rear marches to the end.

● Circular

Pointers **wrap around** with `(rear+1) MOD maxSize`
→ reuses space. Great for buffers.

FINAL PASS BEFORE THE EXAM

Rapid exam tips

Six slips on stack/queue questions.

01

Stack = **LIFO** (top only); queue = **FIFO** (rear in, front out).

02

push: **top++ then store**. pop: **read then top--**.

03

Circular queue wraps with **MOD maxSize** — don't forget it.

04

A **linear** queue wastes freed front space; circular reuses it.

05

Check **isFull** before adding, **isEmpty** before removing.

06

Both are **limited-access** — no reaching the middle.