B2 PROGRAMMING · B2.2

Data structures

Organised ways to store data: **static** vs **dynamic**, **arrays** and **lists**, and the two ordered structures **stacks** (LIFO) and **queues** (FIFO).

01 Key terms

Data structure	Organised way to store data.
Static	Fixed size, set when created.
Dynamic	Grows and shrinks at runtime.
Element	One item in the structure.
Index	Position number; starts at 0.

02 Arrays & lists

Array	Ordered items reached by index.
First index	0; last is length minus 1.
2D array	A grid: array[row][col].
List	Ordered but dynamic; resizes.
Array vs list	Fixed & fast vs flexible size.

03 The structures at a glance

1

Array / list

Access any item by its index. Array is fixed size; list is dynamic.

2

Stack (LIFO)

Add and remove at the top only. Push on, pop off.

3

Queue (FIFO)

Add at the rear, remove from the front. Enqueue, dequeue.

04 Stack — LIFO

Rule	Last In, First Out.
push	Add an item to the top.
pop	Remove the item from the top.
Picture	A stack of plates.
Used for	Undo, browser back, calls.

05 Queue — FIFO

Rule	First In, First Out.
enqueue	Add an item to the rear.
dequeue	Remove from the front.
Picture	A queue of people.
Used for	Print jobs, task scheduling.

06 Know the difference

Stack vs queue	LIFO, add and remove at the top, versus FIFO, add at the rear and remove at the front.	ORDER
Array vs list	Fixed size with fast index access versus a dynamic structure that resizes as needed.	SIZE
Static vs dynamic	Size fixed when created versus size that can grow and shrink while the program runs.	MEMORY
Operation names	Stacks use push and pop; queues use enqueue and dequeue. Use the precise terms.	TERMS

07 Trace tables · the same five operations on each structure

Stack (LIFO) · push(6), push(1), push(9), pop(), push(4)

Operation	Stack (bottom→top)	Returns
push(6)	[6]	-
push(1)	[6, 1]	-
push(9)	[6, 1, 9]	-
pop()	[6, 1]	**9**
push(4)	[6, 1, 4]	-

Queue (FIFO) · enqueue A, B, C, dequeue(), enqueue D

Operation	Queue (front→rear)	Returns
enqueue("A")	[A]	-
enqueue("B")	[A, B]	-
enqueue("C")	[A, B, C]	-
dequeue()	[B, C]	**A**
enqueue("D")	[B, C, D]	-

The contrast in one line: same five operations, but pop() returns the **newest** item (9) while dequeue() returns the **oldest** (A). Always state the returned value, not just the final contents.

Your turn · trace on an empty stack: push(3), push(7), pop(), push(5), pop()

Answer: pop() returns **7**, then pop() returns **5**; the stack finishes as [3].

FINAL PASS BEFORE THE EXAM

Rapid exam tips

Eight things that lose marks if you slip on them. Skim before you walk in.

01

Array indices start at **0**; the last item of n items is at index **n minus 1**.

02

Stack = LIFO (plates), **queue = FIFO** (line of people). Anchor each to its picture.

03

Stacks use **push** and **pop**; queues use **enqueue** and **dequeue**. Use exact terms.

04

A classic **array is static** (fixed size); a **list is dynamic** (resizes).

05

A **2D array** is a grid; reach a cell with **array[row][col]**.

06

Need the **most recent** item first? Use a stack. Need **arrival order**? Use a queue.

07

Reaching past the end is an **out-of-bounds** error; check your loop limits.

08

Match the **structure to the behaviour** the question describes; that earns the marks.