QUEUE QUESTION

1. Practical (Rwanda): At CHUK, 10 patients queue. After 7 served, who is front?

Action	Queue (Front → Back)	Front Patient
Initial State	P1, P2, P3, P4, P5, P6, P7, P8, P9, P10	P1
After 7 served	P8, P9, P10	P8

2. **Practical (Rwanda): At BK ATM, 8 clients queue. Who is third?**

Position	Client ID
1st (Front)	C1
2nd	C2
3rd	C3

Challenge Question

Question: Queue vs stack for canteen meals. Which fits?

The best fit for managing canteen meals (where people are served in the order they arrived) is a **Queue**.

Solution

Step	Algorithmic Sequence	Explanation	Corresponding Code Concept (Python)
1.	**Determine Requirement**	The requirement for a canteen is that the first person who arrives must be the first person to be served.	**FIFO** (First-In, First-Out)

Step	Algorithmic Sequence	Explanation	Corresponding Code Concept (Python)
2.	**Identify Structure for Arrival**	When a client arrives, they are added to the *end* of the line.	**Enqueue**
3.	**Identify Structure for Service**	When a client is served, they are removed from the *front* of the line.	**Dequeue**
4.	**Compare to Stack**	A **Stack** uses **LIFO** (Last-In, First-Out), meaning the last person to arrive would be served first. This is inappropriate for fairness in a meal line.	Stack: list.append() for push, list.pop()
5.	**Conclusion**	The **Queue** structure perfectly models the real-world requirement of a fair service line (FIFO).	Use a **Queue** data structure.

Reflection Question

Question: Why FIFO give fairness in services?

The **First-In, First-Out (FIFO)** principle gives fairness in services because it strictly adheres to the principle of **arrival order**. This mechanism ensures that the element (patient, client, task, etc.) that has waited the longest is the next one to be processed or served.

1. **Impartiality:** FIFO is based on a single, objective metric: **time of arrival**. It does not involve any judgment, priority, or bias based on the size, nature, or value of the item in the queue. Everyone is treated equally based on when they entered the system.
2. **Predictability and Transparency:** Users or processes can clearly understand the system's logic: "If I arrive now, I will be served after all those who are currently ahead of me." This predictability is essential for managing expectations and reducing frustration.

3. **No Starvation:** In systems where elements arrive continuously, other scheduling policies (like priority queuing) can lead to **starvation**, where low-priority elements may never get served. Because FIFO is chronological, every element is guaranteed to eventually reach the front of the queue and be processed.