

3. Challenge Question

Task: Push ["X", "Y", "Z"], pop twice, then push "W". Find the top.

Algorithm (Step-by-step):

1. Start with empty stack.

2. Push X → [X]

3. Push Y → [X, Y]

4. Push Z → [X, Y, Z]

5. Pop twice → [X]

6. Push W → [X, W]

7. Top = "W" ✓

Code:

```
stack = []  
stack.extend(["X", "Y", "Z"])  
stack.pop()  
stack.pop()  
stack.append("W")  
print("Top element is:", stack[-1])
```

Output:

Top element is: W

4. Reflection (Theory – no code)

A stack uses Last In, First Out (LIFO) order.

This is not good for serving people in real life (e.g., bank, hospital, MoMo agent) because the last person gets served first while those who came earlier wait unfairly.

Part B: Queue Questions

1. Practical (Rwanda): RRA office

Task: 4 people join queue. After 2 are served, who is front?

Code:

```
from collections import deque
```

```
queue = deque(["P1", "P2", "P3", "P4"])
```

```
print("Queue:", queue)
```

```
# Serve two
```

```
queue.popleft()
```

```
queue.popleft()
```

```
print("Front person now:", queue[0])
```

Output:

```
Queue: deque(['P1', 'P2', 'P3', 'P4'])
```

```
Front person now: P3
```

2. Practical (Rwanda): Nyabugogo buses

Task: 3 buses enqueue. After 1 departs, which bus is next?

Code:

```
queue = deque(["Bus1", "Bus2", "Bus3"])
```

```
print("Queue:", queue)
```

```
# First bus departs
```

```
queue.popleft()
```

```
print("Next bus:", queue[0])
```

Output:

```
Queue: deque(['Bus1', 'Bus2', 'Bus3'])
```

```
Next bus: Bus2
```

3. Challenge Question (Clinic system)

Algorithm:

1. Patients arrive and join a queue.
2. Each patient is served in order of arrival.
3. If a stack was used instead, last patient would be treated first.

Problem:

This causes unfairness – sick patients who came earlier might be ignored, which is dangerous in a clinic.

4. Reflection (Theory – no code)

Queue (FIFO) = First In, First Out → fairness, order of arrival is respected

Stack (LIFO) = Last In, First Out → unfair in services, only good for temporary tasks (like undo, redo, MoMo steps).