Part I – STACK

A. Basics

1. Operation: Push/Pop (LIFO)
A stack follows the Last In, First Out (LIFO) principle. This means the last item added (pushed) is the first one removed (popped).

Example: In the MTN MoMo app, when you are filling out payment details step-by-step, pressing the “back” button removes the most recent step you completed.

Q1: How does this show the LIFO nature of stacks?
Answer: It shows LIFO because the last step you entered (the most recent action) is the first one removed when you go back, just like the top of the stack is popped first.

2. Operation: Pop (Undo)
Pop removes the top item from the stack. This is useful for undo operations, where the most recent change needs to be reversed.

Example: In UR Canvas, when you go back through course modules or undo an action, the last module or action is undone first.

Q2: Why is this action similar to popping from a stack?
Answer: It is similar because the most recent module you opened or the last action you took is the first to be removed, exactly like popping the top element in a stack.

B. Application

C.

3. Operation: Push (Add to Stack)
Every new transaction or action can be stored by pushing it to a stack, making it easy to reverse later.

Example: In BK Mobile Banking, each transaction is added to a list — this list can be viewed as a stack where the most recent one is on top.

Q3: How could a stack enable the undo function when correcting mistakes?
Answer: The undo function pops the most recent action (like a mistaken transaction) off the stack, allowing the user to go back to the previous state step-by-step.

4. Operation: Balanced Parentheses Check
Stacks can be used to check if parentheses (or form inputs) are properly matched — push opening symbols, pop when a closing one is encountered.

Example: In Irembo registration forms, fields must be filled in matched pairs (e.g., opening and closing form sections).

Q4: How can stacks ensure forms are correctly balanced?
Answer: Each opening form element is pushed onto the stack. When a closing element is found, the top of the stack is popped. If at the end the stack is empty, all fields are properly matched.

C. Logical Reasoning

5. Operation: Push and Pop sequence
Stack actions:

- Push(“CBE notes”)
- Push(“Math revision”)
- Push(“Debate”)
- Pop() → removes “Debate”
- Push(“Group assignment”)

Q5: Which task is next (top of stack)?
Answer: “Group assignment” is now at the top of the stack, ready to be accessed or removed next.

---

6. Operation: Undo with multiple Pops
Suppose a student performs 5 actions and then undoes 3 during an ICT exam.

Q6: Which answers remain in the stack after undoing?
Answer: The first two actions (bottom of the stack) remain, as the top three actions were popped (undone). The stack now contains only the earliest actions.

D. Advanced Thinking

7. Operation: Pop to backtrack
Example: In RwandAir booking, if a user goes back step-by-step to change flight details, each "back" action pops the last step.

Q7: How does a stack enable this retracing process?
Answer: The booking steps are stored in a stack. Going back means popping each recent step one by one, retracing your actions.

8. Operation: Push words, then Pop to reverse
Sentence: “Umwana ni umutware”

- Push: “Umwana” → “ni” → “umutware”
- Pop: “umutware” → “ni” → “Umwana”

Q8: Show how a stack algorithm reverses the proverb.
Answer: By pushing each word onto the stack and then popping them in reverse order, we get: “umutware ni Umwana”.

---

9. Operation: DFS using a stack
Example: A student searches deeply through bookshelves at the Kigali Public Library.

Q9: Why does a stack suit this case better than a queue?
Answer: DFS (Depth-First Search) explores one branch deeply before backtracking. A stack supports this by tracking the path and popping back when a branch ends, unlike a queue which uses BFS (Breadth-First).

---

10. Operation: Push/Pop for Navigation
Example: In BK Mobile app, viewing transaction history uses push to add each new page and pop to go back.

Q10: Suggest a feature using stacks for transaction navigation.
Answer: A “Recent Activity Stack” that allows users to undo the last few navigations or actions within their banking history, retracing steps like in a browser.

Part II – QUEUE

A. Basics

Q1: How does a restaurant queue in Kigali show FIFO behavior?

In a Kigali restaurant, customers are served in the order they arrive. This follows the FIFO (First In, First Out) principle. The first person to enter the queue is the first to be served. This ensures fairness and order, so that no one skips ahead.

- Operation used:
  - Enqueue: Customers join the line at the rear.
  - Dequeue: Customers are served and leave from the front.

This mirrors a FIFO data structure, where elements are processed in the exact order they were added.

Q2: Why is a YouTube playlist like a dequeue operation?

In a YouTube playlist, the next video plays automatically after the previous one ends. This is similar to the dequeue operation where the front item is removed (played) and the next one becomes the new front.

- Operation used:
  - Only dequeue (automatic removal from the front).

Unlike normal queues, there's usually no enqueueing once the playlist starts. This reflects the linear consumption of items from a queue.

B. Application

Q3: How is the line at RRA a real-life queue?

At RRA (Rwanda Revenue Authority), people queue up to pay taxes. This is a clear example of a real-life queue.

- Enqueue: People join the line as they arrive.
- Dequeue: The officer serves the person at the front.

This ensures order, and people are served on a first-come-first-served basis, matching the logic of a queue data structure.

Q4: How do queues improve customer service in MTN/Airtel centers?

Queues ensure that every customer is served in turn, creating a fair and efficient system. At MTN/Airtel service centers, SIM replacement or service requests are handled based on arrival order.

Benefits:

- Reduces confusion and disputes.
- Tracks who should be served next.
- Ensures faster, more reliable service.

This makes operations smooth and gives a better experience to all customers.

C. Logical

Q5: Equity Bank sequence: Enqueue("Alice"), Enqueue("Eric"), Enqueue("Chantal"), Dequeue(), Enqueue("Jean")

Who is at the front now?

Let's process this step by step:

1. Enqueue "Alice" → [Alice]
2. Enqueue "Eric" → [Alice, Eric]
3. Enqueue "Chantal" → [Alice, Eric, Chantal]
4. Dequeue() → Removes "Alice" → [Eric, Chantal]
5. Enqueue "Jean" → [Eric, Chantal, Jean]

✅ Answer:
Eric is now at the front of the queue.

Q6: How does a queue ensure fairness in RSSB pension applications?

RSSB handles pension applications based on arrival time. This guarantees fairness:

- First applicant is served first (FIFO).
- No one is allowed to skip ahead.
- Everyone waits their turn.

Queues prevent favoritism and promote equality in service delivery.

D. Advanced Thinking

Q7: Real-life examples of queue types

- Linear Queue:
  - At a wedding buffet in Rwanda, guests form a straight line.
  - They take food one by one, first in line gets served first.
- Circular Queue:
  - Buses looping at Nyabugogo follow a circular queue.
  - After dropping passengers, the bus rejoins the end of the queue.
- Deque (Double-Ended Queue):
  - In some buses, people can board from both front and rear doors.
  - This allows insertion/removal from both ends.

These examples show how different queue types solve real-world problems.

Q8: How can queues model food ordering in a Kigali restaurant?

At restaurants, customers place orders (enqueue) and are called when food is ready (dequeue). This models:

- Enqueue: Order is placed and added to the kitchen's list.
- Dequeue: When food is ready, the customer is called.

It ensures orders are prepared in the order they were received, creating a fair and manageable system.

Q9: Why is CHUK’s emergency line a priority queue?

In CHUK hospital:

- Emergencies (e.g., accident victims) jump the line.
- Even if they arrived later, they are treated before others.

This is a priority queue, where:

- Each person has a priority level.
- Higher priority (emergencies) are served before lower priority ones.

This ensures critical patients are attended to first, even if it breaks FIFO.

Q10: How would a queue fairly match moto/e-bike drivers and students?

In a moto/e-bike taxi app:

- Enqueue: Students request rides.
- Drivers are matched to students in the order they requested, unless there's a priority.

This ensures:

- First-come, first-served matching.
- No student is skipped.
- Drivers are also assigned fairly.