STACKS & QUEUES QUESTIONS AND ANSWERS

Part I – STACK

A. Basics

Q1: How does this show the LIFO nature of stacks?
In the MTN Momo app, each time you enter details like the phone number, amount, and PIN, those actions are stored one on top of the other. When you press "Back," the last action (PIN entry) is removed first before earlier ones. This shows the **Last In, First Out (LIFO)** nature of a stack, where the most recent entry is always the first one removed.

Q2: Why is this action similar to popping from a stack?
When using UR Canvas and navigating through course modules, pressing the "Back" button undoes the **last navigation step** you took. In stack terms, this is the **Pop operation**: removing the item that is at the top (the most recent step). Just like a stack, you can only remove the last entered action.

B. Application

Q3: How could a stack enable the undo function when correcting mistakes?
When typing or performing transactions in apps like BK Mobile, each action (such as typing a number or confirming a transfer) is **pushed** onto a stack. If a mistake happens, the system can simply **pop** the last action, removing it while keeping earlier actions intact. This is how "Undo" works—removing recent steps one at a time.

Q4: How can stacks ensure forms are correctly balanced?
In Irembo registration forms, when filling data fields with conditions like brackets, parentheses, or tags, each **opening element** is pushed into the stack. When the corresponding **closing element** is found, it pops the opening one. If at the end all items match correctly, the form is balanced. If something is left unmatched in the stack, the form is incomplete or incorrect.

C. Logical

Q5: Which task is next (top of stack)?
Sequence: Push ("CBE notes"), Push ("Math revision"), Push("Debate"), Pop (), Push ("Group assignment").

- After pushing "Debate," the stack is: [CBE notes, Math revision, Debate].
- Pop removes "Debate."

- Push “Group assignment” makes the stack: [CBE notes, Math revision, Group assignment].
  The **top of stack = “Group assignment.”**

Q6: Which answers remain in the stack after undoing?
If during ICT exams a student undoes 3 recent actions, the last three pushed items are popped one by one. What remains in the stack are the earlier actions that were done before the last three. Undo removes only the most recent steps, keeping the first answers safe.

D. Advanced Thinking

Q7: How does a stack enable this retracing process?
In RwandAir booking, when a passenger moves step-by-step and then wants to go back, each booking step is stored in a stack. Pressing “Back” simply pops the top step, taking the passenger back to the previous state. This retracing is possible because of the **LIFO structure**—undoing steps in reverse order.

Q8: Show how a stack algorithm reverses the proverb “Umwana ni umutware.”

- Push: [Umwana], then [ni], then [umutware].
- Stack becomes: [Umwana, ni, umutware] (with “umutware” at the top).
- Pop sequence: First = “umutware,” then “ni,” then “Umwana.”
  Reversed proverb: **“Umutware ni Umwana.”**

Q9: Why does a stack suit this case better than a queue?
In DFS (Depth-First Search), like searching bookshelves in Kigali Public Library, you go **deep into one shelf before moving back.** This requires remembering the last shelf visited to backtrack, which is perfectly handled by a stack. A queue would not work well because it follows breadth-first (wide search) instead of depth-first.

Q10: Suggest a feature using stacks for transaction navigation.
In BK Mobile app, a stack can be used so that when a user views transactions, each opened transaction detail is pushed. If they want to go back, they pop step by step. A possible feature: **“Step-back transaction history”** where you can retrace previously opened transactions in reverse order without losing track.

Part II – QUEUE

A. Basics

Q1: How does this show FIFO behavior?
In a Kigali restaurant, customers are served in order they arrive. The **first customer to enter**

is the first to be served, and the **last to arrive** waits at the end. This is exactly **FIFO (First In, First Out)**.

Q2: Why is this like a dequeue operation?
In a YouTube playlist, the next video automatically plays after the current one finishes. This is like **dequeue**, where the first video in the list is removed (played), and the next one in line becomes the new front.

B. Application

Q3: How is this a real-life queue?
At RRA offices, people line up to pay taxes. The first taxpayer in the queue is served first, and the next waits for their turn. This mirrors the **Enqueue/Dequeue process** of a queue.

Q4: How do queues improve customer service?
In MTN/Airtel centers, people with SIM replacement requests are handled one after another. A queue ensures **fairness, order, and efficiency**, avoiding confusion or fights. Customers know they will be served in the order they arrived.

C. Logical

Q5: Who is at the front now?
Sequence: Enqueue("Alice"), Enqueue("Eric"), Enqueue("Chantal"), Dequeue (), Enqueue("Jean").

- Queue after three enqueues: [Alice, Eric, Chantal].
- Dequeue removes "Alice."
- Enqueue "Jean": [Eric, Chantal, Jean].
  The **front = Eric.**

Q6: Explain how a queue ensures fairness.
In RSSB pension applications, requests are handled in the order they arrive. A queue prevents skipping or cheating. Everyone is treated equally because the first application submitted is processed first, ensuring **transparency and fairness.**

D. Advanced Thinking

Q7: Explain how each map to real Rwandan life.

- **Linear queue**: At a wedding buffet, people line up once, and each gets served only once.

- **Circular queue**: At Nyabugogo bus station, buses operate in cycles, going around and coming back to the starting point.
- **Deque (Double-ended queue)**: Boarding a bus where some passengers enter from the front door and others from the rear door—both ends are active.

Q8: How can queues model this process?
In a Kigali restaurant, when customers order food, each order is **enqueued**. The kitchen prepares them in order, and when ready, they are **dequeued** as customers are served. This keeps preparation and serving organized.

Q9: Why is this a priority queue, not a normal queue?
At CHUK hospital, emergencies are treated before ordinary patients. Unlike a normal queue (FIFO), a **priority queue** serves based on urgency. Critical patients “jump the line,” showing it is not strict FIFO.

Q10: How would queues fairly match drivers and students?
In a moto/e-bike taxi app, passengers (students) and drivers are each placed in queues. The app **dequeues** the first driver and the first student to match them fairly. This prevents someone who came later from being served earlier, ensuring fairness and order.