

NAMES: BIZIMUNGU JAMES

REG No: 224016991

DEPARTMENT: BUSINESS INFORMATION AND TECHNOLOGY

YEAR OF STUDY: YEAR 2 BIT- GROUP 1

ASSIGNMENT: DATA STRUCTURE AND ALGORITHM

Q1. MTN MoMo (LIFO)?

Pressing **back** removes the *last* step you completed. That mirrors LIFO: the most recently pushed item is the 5 first popped — last input → first undone.

Q2. UR Canvas (pop)?

Pressing back removes the last visited page/state — exactly like a **pop** operation that removes the top element from the stack of visited pages.

B. Application

Q3. BK Mobile: undo via stack

Store each user action (form edit, transaction entry, etc.) as a **push** onto an actions stack. To undo, **pop** the stack and revert that action. This naturally reverses actions in the correct (most-recent-first) order.

Q4. Balanced fields (parentheses check)

When an opening field/marker is encountered push it. When a corresponding closing marker is seen, check the top of the stack and **pop** if it matches. At the end the stack must be empty — otherwise there are unmatched/open fields. This prevents mismatch/errors in nested inputs.

C. Logical

Q5. Given sequence

Sequence: Push ("CBE notes"), Push("Math revision"), Push("Debate"), Pop(), Push("Group assignment")

- After the Pop() the “Debate” entry is removed. Then “Group assignment” is pushed.

Top of stack = "Group assignment".

Q6. Undo 3 recent actions

If the student performs three pop() operations on the current stack, the three most recent items are removed.

assignment), undoing 3 removes them all.

Remaining = none (stack empty).

D. Advanced Thinking

Q7. Backtracking in RwandAir booking

Each completed step (choose flight, passenger info, seats, payment info) is **pushed**. When the passenger goes back step-by-step, the system **pops** each top state and restores the previous state. The stack stores the trail, enabling exact stepwise retracing.

Q8. Reverse “Umwana ni umutware” using a stack

Algorithm (word-level):

- Input words in order: ["Umwana", "ni", "umutware"]
- Push each: push("Umwana"), push("ni"), push("umutware")
Stack top→bottom: umutware, ni, Umwana
- Pop until empty and output popped words: pop→umutware, pop→ni, pop→Umwana
Result (reversed): **"umutware ni Umwana"**.

Q9. DFS (stack) vs queue for searching shelves

DFS explores one branch deeply before backtracking — exactly what a stack (LIFO) does: push neighbors, always visit the most recently discovered next shelf, and pop to backtrack. A queue (FIFO) would do breadth-first (visit all neighbors shallowly) which is less suitable when you want to follow one path deeply (e.g., search inside a shelf then its sub-shelves).

Q10. Feature suggestion for BK Mobile (using stacks)

Undo/Redo browsing in transaction history:

- Maintain two stacks: a **Back** stack and a **Forward** stack.
- When the user views a transaction, push current view to **Back**.
- Press “Back” → pop Back (go to previous) and push the popped view onto **Forward**.

A. Basics

Q1. How does this show FIFO behavior?

Customers are served in arrival order: first to arrive is first to be served. This is **First-In, First-Out (FIFO)** — the queue data structure model.

Q2. YouTube playlist (dequeue)

The playlist plays the next video (front of the queue) and removes it when done — same as **dequeue**: remove from front.

B. Application

Q3. RRA tax line

People arriving wait in line (enqueue at rear); the clerk processes the earliest waiting person first (dequeue from front). It's a direct real-life queue.

Q4. SIM replacement queue (improved service)

Queues keep order and transparency, reduce conflicts, let staff process customers fairly and predictably, and allow metrics (wait times) so resources can be allocated (add more servers at peak).

C. Logical

Q5. Equity Bank sequence

Operations: Enqueue("Alice"), Enqueue("Eric"), Enqueue("Chantal"), Dequeue(), Enqueue("Jean")

- Dequeue() removes "Alice". Queue front now is "Eric". After Enqueue("Jean"), queue (front→rear) = Eric, Chantal, Jean.

Front = "Eric".

Q6. FIFO fairness (RSSB pensions)

A queue ensures fairness because applicants are served strictly by arrival order—no one who arrived later is served before someone earlier — first come, first served.

D. Advanced Thinking

Q7. Queue types mapped to Rwandan life

- **Linear queue**: people lining once at a wedding buffet — once served you leave (no reuse of space).
(producers and consumers wrap around).

- **Deque (double-ended):** boarding a bus that allows entry from front and rear (you can enqueue/dequeue from either end).

Q8. Restaurant orders modeled by queues

Customers place orders (enqueue to an order queue). Kitchen prepares in FIFO; when ready the order is dequeued and the customer is notified/called. This ensures older orders are prioritized and reduces mix-ups.

Q9. Priority queue at CHUK

Emergencies are assigned higher priority and served before lower-priority patients, even if they arrived later. That's a **priority queue** — elements with higher priority are dequeued before lower-priority ones (not pure FIFO).

Q10. Fair matching in a moto/e-bike taxi app

Use queues for riders and drivers with fairness rules:

- Maintain a FIFO rider queue per zone; drivers dequeue from the front when available.
- To balance fairness and efficiency, add priority factors (e.g., driver proximity, driver rating) or time-based aging (increase priority the longer a rider waits).

This preserves fairness while keeping service efficient