UNIVERSITY OF RWANDA
BIT LEVEL TWO
DATA STRUCTURE AND ALGORITHMS ASSIGNMENT
NAMES:MANISHIMWE Jean Bosco
Registration number:224017179

BIT - Data structure Exercise - number 2

Part I- STACK

A. Basics

1. Operation: Push/Pop (LIFO) — In a stack, the last item added is the first removed.

In the MTN MoMo app, when you fill payment details step-by-step, pressing back removes the last step.

Q1: How does this show the LIFO nature of stacks?

2. Operation: Pop (Undo) — Pop removes the top item.

   In UR Canvas, when you navigate course modules, pressing back undoes the last step.

   Q2: Why is this action similar to popping from a stack?

B. Application

3. Operation: Push (Add to stack) — New actions are added to the stack top. In BK Mobile Banking, transactions are added to history.

   Q3: How could a stack enable the undo function when correcting mistakes?

4. Operation: Balanced Parentheses Check (Stack-based matching) — Push opening bracket, pop when matching closing bracket is found.

   In Irembo registration forms, data entry fields must be correctly matched. Q4: How can stacks ensure forms are correctly balanced?

C. Logical

5. Operation: Push and Pop sequence.

   A student records tasks in a stack:

   Push("CBE notes"), Push("Math revision"), Push("Debate"), Pop(),

   Push("Group assignment")

   Q5: Which task is next (top of stack)?

6. Operation: Undo with multiple Pops.

   During ICT exams, a student undoes 3 recent actions.

   Q6: Which answers remain in the stack after undoing?

D. Advanced Thinking

7. Operation: Pop to backtrack.

   In RwandAir booking, a passenger goes back step-by-step in the form. Q7: How does a stack enable this retracing process?

8. Operation: Push words, then Pop to reverse.

   To reverse "Umwana ni umutware", push each word and then pop. Q8: Show how a stack algorithm reverses the proverb.

9. Operation: DFS using a stack.

   A student searches shelves in Kigali Public Library (deep search). Q9: Why does a stack suit this case better than a queue?

10.Operation: Push/Pop for navigation.

In BK Mobile app, moving through transaction history uses push and pop. Q10: Suggest a feature using stacks for transaction navigation.

Part II- QUEUE

A. Basics

1. Operation: Enqueue (add at rear), Dequeue (remove from front).

   At a restaurant in Kigali, customers are served in order. Q1: How does this show FIFO behavior?

2. Operation: Dequeue (next item leaves first).

   In a YouTube playlist, the next video plays automatically.

   Q2: Why is this like a dequeue operation?

B. Application

3. Operation: Enqueue (job submission).

   At RRA offices, people waiting to pay taxes form a line. Q3: How is this a real-life queue?

4. Operation: Queue management.

   In MTN/Airtel service centers, SIM replacement requests are processed in order.

   Q4: How do queues improve customer service?

C. Logical

5. Operation: Sequence of Enqueue/Dequeue.

   In Equity Bank, operations are:

   Enqueue(“Alice”), Enqueue(“Eric”), Enqueue(“Chantal”), Dequeue(), Enqueue(“Jean”)

   Q5: Who is at the front now?

6. Operation: FIFO message handling.

   RSSB pension applications are handled by arrival order.

   Q6: Explain how a queue ensures fairness.

D. Advanced Thinking

7. Operation: Different queue types. Examples:
    - Linear queue = people at a wedding buffet.
    - Circular queue = buses looping at Nyabugogo.
    - Deque = boarding a bus from front/rear.

    Q7: Explain how each maps to real Rwandan life.
8. Operation: Enqueue orders, Dequeue when ready.

    At a Kigali restaurant, customers order food and are called when ready. Q8: How can queues model this process?
9. Operation: Priority queue.

    At CHUK hospital, emergencies jump the line.

    Q9: Why is this a priority queue, not a normal queue?
10. Operation: Enqueue/Dequeue matching system.

    In a moto/e-bike taxi app, riders wait for passengers.

    Q10: How would queues fairly match drivers and students?

------ END ----

ANSWERS

Part I:STACK

A. Basics

Ans1. MTN MoMo (LIFO)?

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

Ans2. 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 Ans3. 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.

Ans4.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 Ans5. 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".**

Ans6 .Undo 3 recent actions

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

- If applied to the stack state after Q5 (which from bottom→top is: CBE notes, Math revision, Group assignment), undoing 3 removes them all. **Remaining = none (stack empty).**

D. Advanced Thinking Ans7 .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.

Ans8. 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"**.

Ans9. 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).

Ans10. 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**.
- Press "Forward" → pop Forward and push it back to Back.
  This gives browser-like navigation and a safe undo/redo for viewed transactions.

<u>Part II :QUEUE</u>

A. Basics Ans1. Restaurant FIFO

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

Ans2. 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 Ans3.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.

Ans4.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 Ans5 .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".**

Ans6. 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 Ans7. Queue types mapped to Rwandan life

- **Linear queue:** people lining once at a wedding buffet — once served you leave (no reuse of space).
- **Circular queue:** buses looping at Nyabugogo — spots reuse as buses re-enter the loop (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).

Ans8. 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.

Ans9. 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).

Ans10 .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.