ASSIGNEMENT OF DATA BASE STRUCURE AND ALLOGRATIM

Manishimwe Emile

RegN: 224010949

A. BASICS

Q1. When we use MTN MOMO app and enter payment details

- You push each step like amount, PIN, and confirmation onto the stock one after another.
- If you press back the app removes the most recent step you entered confirmation first, then PIN then amount

This shows the Last in First Out (LIFO) principle of stack because:

- The last detail you entered confirmation is the first one removed when press back.
- You cannot go directly to first step amount without first the later steps.

So, the MOMO app's back bu7tton behavior mirrors exactly how a stack in computer science: Push =going forward - step by step and Pop=going backward- step by step (removing the last step.

Q2. This action is like popping from stock because both follow the Last in First Out (LIFO) principle. Here`s how they relate:

Stack Behavior

A stack is data structure where the last item added is the first removed, the pop operation removes the topmost item   the most recent addition.

UR Canvas Navigation: When you navigate through course module each step like opening a module or page is added to history stack.

Pressing Back undoes the last navigation step just like popping the top item off a stack.

Why it is Similar. The most recent action like opening module is undone first exactly how a stack removes the last item off a stack. This ensures a natural and intuitive undo experience mirroring how stacks manage order.

B. APLICATION

Q3.A stack works on the Last in First Out principle. The most recent action is the first one to be undone just like you would remove the top item from a stack of plates likes: You are using BK Mobile Banking: Transfer money to a friend, pay for your electricity, Buy airtime.

Each of these actions is pushed onto a stack, the most recent one buying airtime is at the top.

Now suppose you realize you made a mistake buying airtime you sent it wrong.

Undo with Stack

To correct the mistake:

The system pops the top item from the stack to the airtime purchase.

It undoes that transaction if allowed and now the electricity payment I s at the top.

You can continue popping if you want to undo more steps.

Why this work

Push adds each new action to the top of the stock.

Pop removes the most recent action perfect for undoing mistakes in reverse order.

This keeps history clean and lets you backtrack safely.

Q4. Every opening symbol (, {, [) must have a matching closing symbol (,},]) in the correct order.

A stack helps track these symbols because it works in LIFO (Last In, First Out) order.

Application to Irembo Forms

In Irembo registration forms, every field (like *Name*, *ID Number*, *Date of Birth*, etc.) must be correctly filled and closed before submission.

- If you "open" a field (start filling it in), it should be properly closed/validated (completed correctly), A stack can be used to keep track of open fields and when you start filling it, it is pushed onto the stack and when you finish filling and validate, it is popped. If at the end the stack is empty → all fields were properly matched and completed, and if not → some fields are incomplete or mismatched. Example: Name, ID Number, Date of Birth

Process using a stack:

Start filling Name → Push, Complete Name → Pop, Start filling ID Number → Push and Complete ID Number → Pop.

Start filling Date of Birth → Push Complete Date of Birth → Pop) At the end → Stack is empty → Form is correctly balanced.

Error Example If the user starts filling fields but does not complete one:

- Push for Name → Pop and Push for ID Number → user leaves it incomplete.

At submission, stack is not empty → form is unbalanced → system shows error: Please *complete all required fields.*

C.LOGICAL

Q5. Logic→ Stack = CBE notes, Math revision
Top = *Math revision*

Push: Group assignment
Stack = CBE notes, Math revision, Group assignment
Top = *Group assignment*

Q6. In a stack, "Undo" = Pop (remove the most recent actions).
If the student undoes 3 recent actions, which means we perform 3 Pops.

Example scenario:

Suppose the student's stack of actions is:
["Q1 answer", "Q2 answer", "Q3 answer", "Q4 answer", "Q5 answer"]

- Pop 1 → removes "Q5 answer."
- Pop 2 → removes "Q4 answer."
- Pop 3 → removes "Q3 answer."

Remaining in the stack: ["Q1 answer", "Q2 answer"]

D. ADVANCED THINKING

Q7: Operation – Pop to backtrack (RwandAir booking form)

When booking a ticket online (like in the RwandAir system), each step you complete (e.g., enter passenger name, choose flight, select seat, make payment) is pushed onto a stack.

If the passenger wants to go back step-by-step, the system performs Pop operations to remove the most recent action first. This is because a stack follows LIFO (Last In, First Out) order.

How a stack enables retracing: Each new step is stored at the top of the stack. When the passenger clicks "Back," the system pops the last step (removes it). This continues until the passenger reaches the step they want to edit or review.

Q8: Operation – Push words, then Pop to reverse

We want to reverse the prover. Umwana ni umutwareStep 1: Push each word into the stack.

We read the words one by one and push them:

- Push(“U”) → Stack: [Umwana]
- Push(“ni”) → Stack: [Umwana, ni]
- Push(“umutware”) → Stack: [Umwana, ni, umutware]

Step 2: Pop words from the stack

Now we pop one by one (LIFO order):

- Pop () → “umutware”
- Pop () → “ni”
- Pop () → “Umwana”

Final Reversed Proverb:

umutware ni Umwana

Algorithm in Simple Pseudocode

```
Initialize empty stack.
For each word in "Umwana ni umutware": Push(word)
While stack is not empty:
 word = Pop ()
   Print(word)
```

Q9.DFS (Depth-First Search) means going deep into one path first before backtracking.

A student searching for a book in Kigali Public Library:

- The library has many shelves (like nodes in a graph).
- Each shelf has sections (like edges to other nodes).
- The student decides to go deep into one section fully before checking another.

Why a Stack Fits Better than a Queue

- Stack (LIFO):
    - Last shelf visited is the first to be explored next.

- This allows the student to go deep quickly and backtrack step-by-step (like "Back" in the booking form earlier).

- Queue (FIFO):
    - First shelf I visited was the first explored.
    - This gives a breadth-first search (BFS), checking all nearby shelves first, not going deep.

Since the student wants a deep search (DFS), the stack is better.

Q10. Operation – Push/Pop for navigation (BK Mobile App)

In the BK Mobile app, a customer can check past transactions (like deposits, transfers, airtime purchases).
To navigate forward and backward smoothly, we can use a stack.

Feature: Back and Forward Transaction Navigation, each time the user opens a transaction detail, that screen is pushed onto a stack When the user presses Back, the app pops the last opened transaction and returns to the previous one and This stack-based navigation makes it easy for customers to move step-by-step backward and forward through their transaction history, just like browsing pages in Chrom

PART TWO: QUEUE

A. **BASICS**
Q1. In a queue, the rule is FIFO (First In, First Out) the first item added is the first one removed. Kigali restaurant example: Customers join the line at the rear (Enqueue). The restaurant serves customers from the front (Dequeue)o, the first customer to arrive will also be the first one to be served.

)shows FIFO behavior because the order of service follows the order of arrival — first come, first served.

Q2: Operation – Dequeue in a YouTube Playlist

In a YouTube playlist, videos are arranged in a specific order (like a queue): Front of the queue = the next video to play, When the current video finishes, the next one is removed from the front (like a Dequeue operation) and starts playing, Latest videos you add to the playlist are usually added at the rear.

Why is it like Dequeue:

- Dequeue removes the front item first, In YouTube, the playlist automatically removes/plays the first video in line and shifts to the next. This shows FIFO behavior: the video that entered the playlist earliest (and is at the front) plays first.

B. Application

Q3. Operation – Enqueue (job submission) at RRA offices

At RRA (Rwanda Revenue Authority) offices:

- People arrive to pay taxes and join a line (queue).
- Enqueue (add at rear): Each new person joins the end of the line.
- Dequeue (remove from front): The person at the front of the line is served first.

Why is a real-life queue:

- It follows FIFO (First In, First Out): the first person to arrive is the first to be served.
- Each person waits their turn, just like items in a queue data structure.

Q4. How queues improve customer service at MTN/Airtel service centers

1. Fairness: Customers are served in the order they arrive (FIFO), so no one is skipped.
2. Organization: Staff can process SIM replacement requests systematically, avoiding confusion.
3. Efficiency: The queue ensures smooth flow, helping staff serve one customer at a time.
4. Transparency: Customers can see their place in line and have a clearer idea of waiting time.

C.Logicals

Q5. Sequence of Enqueue/Dequeue at Equity Bank

Operations:

Enqueuer("Alice") → Queue: [Alice]

Enqueue("Eric") → Queue: [Alice, Eric]

Enqueue("Chantal") → Queue: [Alice, Eric, Chantal]

Dequeue () → removes Alice → Queue: [Eric, Chantal]

Enqueue("Jean") → Queue: [Eric, Chantal, Jean]

Front of the queue now: Eric

Q6In **RSSB pension applications**, files are placed in a **queue** — first submitted, first processed.

How a queue ensures fairness:

1. **Arrival order respected** – Whoever submits first is processed first (FIFO).
2. **No skipping** – Later applicants cannot jump ahead of earlier ones.
3. **Equal treatment** – Every application waits its proper turn.
4. **Transparency** – Customers trust the system because it is predictable and orderly.

Therefore, the queue makes the process **fair and organized**, ensuring that all pensioners are treated equally.

C. Advanced thinking

Q7. Linear Queue – People at a wedding buffet

- Guests line up one after another to serve food, The first guest in line eats first, and the last waits longest and This is FIFO (First In, First Out).

  2. Circular Queue – Buses looping at Nyabugogo: Buses complete their trip, then return to the end of the line for the next round, The queue "wraps around," just like a circular queue, ensures continuous rotation without leaving gaps.

  3. Deque (Double-Ended Queue) – Boarding a bus from front/rear: At some bus stations, passengers can enter either the front or the back door and This means people can join or leave from both ends of the line, just like in a deque.

In short: Linear queue = one-way, fair order, Circular queue = endless rotation, no empty slots and Deque = flexibility to use both ends

Q8: How queues model food orders at a Kigali restaurant

At a restaurant: Enqueue (add at rear): When a customer places an order, it is added to the end of the order queue.

Dequeue (remove from front): When the food is ready, the first order in the queue is served first.

Why a queue works:

Ensures FIFO (First In, First Out): the first customer to order is the first to receive their food.

- Prevents confusion or skipping orders.
- Make kitchen workflow organized and predictable.

Q9: Priority Queue at CHUK Hospital

At CHUK hospital:

- Patients arrive and normally wait in line (queue).
- Emergencies are treated before regular patients, even if they arrive later.
- This is not a normal queue because it does not strictly follow FIFO.

Why it is a Priority Queue:

- Priority-based serving: Patients with higher urgency (e.g., emergency cases) are served first.
- Normal queue (FIFO) serves strictly in arrival order; emergencies would have to wait.
- Priority queue allows reordering based on severity/urgency, not just arrival time.

Q10. Queue system in a moto/e-bike taxi app.

In a taxi app:

1. Enqueue (add at rear): Riders who are available join a queue to wait for passenger requests.and Passengers who request a ride are also added to a queue if no driver is immediately available.
2. Dequeue (remove from front): When a passenger requests a ride, the first available rider in the queue is matched. And This ensures the earliest available driver gets assigned first. How this ensures fairness: FIFO principle: Drivers and passengers are matched in the order they become available, prevents favoritism; no driver or passenger is skipped and ensures a smooth and predictable system for both riders and passengers.
