Names: BYIRINGIRO Evode

Reg Number: 224016899

Department: BIT Year Two

Data structure and algorithm assignment

Part I – STACKS

A. Basics

Q1. How does this show the LIFO nature of stacks?
In the MTN MoMo app, the steps you follow are arranged like items in a stack. The last datail you entered is the first one removed when you press back. This is exactly how LIFO works, for example, when you entering data in MTN MOMO app allows you to go back while you're not confirm or execute final instruction so, by backing you reach to other recently data by removing last data to reach recent one

Q2. Why is this action similar to popping from a stack?
On UR Canvas, going back while navigating course modules removes the most recent step. This is the same as the pop operation, where the top element of the stack is taken out, and leaving earlier actions still in place. For example, when we create a list or array and we need to pop it removes last number on list or string that's how similar in real sense

---

B. Application

Q3. How could a stack enable the undo function when correcting mistakes?
Whenever you perform an action in BK Mobile Banking, it is pushed onto the stack. If a mistake occurs, the undo function pops the last action, restoring the previous correct state. This is possible because stacks always remember the most recent step at the top. For example, when entered fund amount in mistake way we cancel that transaction and make reversal.

Q4. How can stacks ensure forms are correctly balanced?
In Irembo forms, each opening field must be closed properly before submitting. Stacks handle this by pushing each opening bracket (or field) and popping it when a matching closing part is found. If the stack is empty at the end, the form is correctly balanced.

C. Logical

Q5. Which task is next (top of stack)?
Sequence: Push (“CBE notes”), Push (“Math revision”), Push (“Debate”), Pop (), Push (“Group assignment”).
After popping “Debate”, the new top becomes “Group assignment”.

Q6. Which answers remain in the stack after undoing?
If a student undoes 3 actions, then the last three pushed items are popped out. The earlier answers remain in the stack, showing only the work done before those three steps.

D. Advanced Thinking

Q7. How does a stack enable this retracing process?
In RwandAir booking, each step you fill is pushed into a stack. When the passenger goes back, the stack pops the latest step, allowing retracing one step at a time until the beginning is reached.

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

- Push (Umwana), Push (Ni), Push (umutware).
- Pop gives: “umutware ni Umwana”.
  Thus, stacks reverse the order of words by popping last-in items first.

Q9. Why does a stack suit this case better than a queue?
Searching books deeply in Kigali Public Library follows the DFS method. A stack works best here because it allows going as deep as possible into one shelf before backtracking, unlike a queue which would only search level by level.

Q10. Suggest a feature using stacks for transaction navigation.
In the BK Mobile app, a stack could be used to track transaction steps. Users could press back to pop the last transaction detail, or forward to re-push it, making navigation smooth and consistent.

Part II – QUEUES

A. Basics

Q1. How does this show FIFO behavior?
At a Kigali restaurant, customers are served in the same order they arrive. The first person in line is also the first to be served, which is the essence of FIFO.

Q2. Why is this like a dequeue operation?
In a YouTube playlist, the first video in the queue plays first. Removing that video and moving to the next is similar to dequeuing the front element of a queue. The very first video to be queued is also the first one to play. Once it finishes, it is removed, and the next video automatically starts. This is the same as a dequeue operation, where the item at the front of the queue is taken out first.

B. Application

Q3. How is this a real-life queue?
At RRA offices, taxpayers line up and wait their turn. Each person is enqueued at the back and served in sequence. This is a real-world queue because no one skips unless priority rules are applied.

Q4. How do queues improve customer service?
In telecom service centers, SIM replacements are handled in the order requests arrive. Queues prevent confusion, ensure fairness, and reduce conflicts by serving customers step by step.

C. Logical

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

- “Alice” leaves first.
- Now the front is **Eric**.

Q6. Explain how a queue ensures fairness.
RSSB pension applications are handled in the order received. This guarantees fairness because no one can jump the line; everyone waits their turn as in a real queue.

D. Advanced Thinking

Q7. Explain how each maps to real Rwandan life.

- **Linear queue**: People at a wedding buffet, moving forward in a straight line.
- **Circular queue**: Buses at Nyabugogo that loop around to restart their journey.

- **Deque**: Passengers boarding a bus either from the front or the rear door.

Q8. How can queues model this process?
In a Kigali restaurant, when customers order food, their requests are enqueued. When food is ready, the first order placed is dequeued and served, maintaining order.

Q9. Why is this a priority queue, not a normal queue?
At CHUK hospital, emergencies are treated before regular cases. This is a priority queue since critical patients get higher priority than those who came earlier.

Q10. How would queues fairly match drivers and students?
In a moto/e-bike app, available drivers are enqueued while students also join the queue when requesting a ride. The system dequeues one driver and one student at the same time, ensuring fair and timely matching.

Conclusion

In reality Stacks and queues may look abstract in theory, but they shapes real life as they clearly reflect daily life. From banking apps, hospitals, restaurants, to transport systems, these data structures organize tasks in ways that ensure order, fairness, and efficiency.