HUYE CAMPUS

COLLEGE OF BUSINESS AND ECONOMICS

BUSINESS INFORMATION AND TECHNOLOGY

GROUP: 1

DATA STRUCTUTURE AND ALGORITHMS ASSIGNMENT

NO	NAMES	REG NUNBER
114	UWANDOYENEZA ODILLE	224014500

Part I – STACK

A. Basics

Q1: How does this show the LIFO nature of stacks?
☞ In the MTN MoMo app, each payment detail step is pushed onto the stack. When you press back, the most recent step (last entered) is removed first. This is exactly Last-In, First-Out (LIFO).

Q2: Why is this action similar to popping from a stack?
☞ In UR Canvas, pressing back removes the most recent navigation step. This mimics the pop operation, where the top item is removed, undoing the last action.

B. Application

Q3: How could a stack enable the undo function when correcting mistakes?
☞ Every user action is pushed onto the stack. If a mistake happens, the undo command simply pops the last action, restoring the state before the mistake.

Q4: How can stacks ensure forms are correctly balanced?
☞ When filling Irembo forms, each opening field/entry (like "start of section") is pushed. Each closing field (like "end of section") must match by popping. If all match correctly, the form is balanced; otherwise, it's invalid.

C. Logical

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

- Push("CBE notes") → [CBE notes]
- Push("Math revision") → [CBE notes, Math revision]
- Push("Debate") → [CBE notes, Math revision, Debate]
- Pop() → removes "Debate" → [CBE notes, Math revision]
- Push("Group assignment") → [CBE notes, Math revision, Group assignment]
  ☞ Top of stack = Group assignment.

Q6: Which answers remain in the stack after undoing?
☞ Undo = 3 pops. If the stack was [A1, A2, A3, A4, A5] (top = A5):

- Pop A5 → [A1, A2, A3, A4]

- Pop A4 → [A1, A2, A3]
- Pop A3 → [A1, A2]
  ☞ Remaining = A1, A2.

D. Advanced Thinking

Q7: How does a stack enable this retracing process?
☞ In RwandAir booking, each step is pushed as you move forward. To go back, you pop steps one by one, retracing in reverse order.

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

- Push words → [Umwana, ni, umutware]
- Pop → “umutware”
- Pop → “ni”
- Pop → “Umwana”
  ☞ Reversed = “umutware ni Umwana”.

Q9: Why does a stack suit this case better than a queue?
☞ In library DFS, you must go deep into one shelf before backtracking. A stack supports this because it always goes back to the most recent branch (LIFO). A queue (FIFO) would spread search level by level (BFS).

Q10: Suggest a feature using stacks for transaction navigation.
☞ Add an “Undo last transaction view” feature: Each screen is pushed; pressing back pops the last screen, letting users retrace their navigation history.

Part II – QUEUE

A. Basics

Q1: How does this show FIFO behavior?
☞ In a Kigali restaurant, the first customer to join the line is served first. This is First-In, First-Out (FIFO).

Q2: Why is this like a dequeue operation?
☞ In YouTube playlists, the front video plays first and is removed, just like dequeue removes from the front.

B. Application

Q3: How is this a real-life queue?

☞ At RRA offices, each taxpayer joins the line (enqueue). The first taxpayer served is the first who joined (dequeue). This is a classic queue.

Q4: How do queues improve customer service?

☞ Queues ensure fairness and order — no skipping, confusion, or overlapping service. Everyone is served based on arrival time.

C. Logical

Q5: Who is at the front now?

☞ Sequence:

- Enqueue(“Alice”) → [Alice]
- Enqueue(“Eric”) → [Alice, Eric]
- Enqueue(“Chantal”) → [Alice, Eric, Chantal]
- Dequeue() → removes “Alice” → [Eric, Chantal]
- Enqueue(“Jean”) → [Eric, Chantal, Jean]
  ☞ Front = Eric.

Q6: Explain how a queue ensures fairness.

☞ In RSSB pension applications, requests are handled in order of arrival (FIFO). Nobody can jump the line; everyone waits their turn fairly.

D. Advanced Thinking

Q7: Explain how each maps to real Rwandan life.

- Linear queue: People lining at a wedding buffet → served in strict arrival order.
- Circular queue: Buses looping at Nyabugogo → when one finishes, it goes back to the end of the loop.
- Deque: Boarding buses from front/rear → people can enter or exit from both ends.

Q8: How can queues model this process?

☞ Customers enqueue orders. When food is ready, it is dequeued (served). The restaurant processes in the same order as received.

Q9: Why is this a priority queue, not a normal queue?
☞ At CHUK hospital, emergencies are served before others, regardless of arrival time. This breaks FIFO rules → making it a priority queue.

Q10: How would queues fairly match drivers and students?
☞ In a moto/e-bike taxi app:

- Riders (drivers) enqueue as they become available.
- Students enqueue as they request rides.
- The system dequeues one driver and one student at the same time, ensuring fair matching without skipping.