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 ----