WIA1002 DATA STRUCTURE
LAB TEST 2
Duration: 1 hour
(Wednesday 4.00pm – 5.00pm)

Question:
You are tasked with developing a ticketing system for a theme park. This system needs to efficiently manage the flow of visitors entering and exiting the park, as well as tracking the waiting lines for various rides.

Instructions:

1. **Entrance Gate Management (Stack):** Implement a stack-based system to manage the entrance gates of the theme park. Follow these steps:
   - Create a Java class named **EntranceGateStack** to represent the stack for gate management.
   - Implement the following functionalities within the **EntranceGateStack** class:
     - **assignGate(gateNumber)**: Pushes the assigned gate number onto the stack when visitors arrive at the park.
       (1 mark)
     - **releaseGate()**: Pops and releases the gate number from the stack when visitors leave the park.
       (1 mark)
     - **checkTopGate()**: Returns the gate number at the top of the stack without removing it.
       (1 mark)
     - **isStackEmpty()**: Returns true if the stack is empty, indicating no gates are occupied.
     - **isStackFull()**: Returns true if the stack is full, indicating all gates are occupied.
       (1 mark)

2. **Ride Waiting Line Management (Queue):** Develop a queue-based system to manage the waiting lines for rides within the theme park. Follow these steps:
   - Create a Java class named **RideWaitingQueue** to represent the queue for managing ride waiting lines.
   - Implement the following functionalities within the **RideWaitingQueue** class:
     - **joinWaitingLine(ticketNumber)**: Enqueues the ticket number into the waiting line when visitors join the line for a ride.
       (1 mark)
     - **completeRide()**: Dequeues and processes the ticket number from the waiting line when visitors complete the ride.
       (1 mark)

- **viewNextTicketNumber()**: Returns the next ticket number in the waiting line without removing it.

(1 mark)

- **isQueueEmpty()**: Returns true if the queue is empty, indicating no visitors are waiting for the ride.
- **isQueueFull()**: Returns true if the queue is full, indicating the waiting line is at maximum capacity.

(1 mark)

3. **Testing and Verification:**
   - Write a **main** method to test the functionality of both the **EntranceGateStack** and **RideWaitingQueue** classes.
   - Simulate visitor arrivals and departures at entrance gates using the **EntranceGateStack** methods.
   - Simulate visitors joining and completing rides using the **RideWaitingQueue** methods.
   - Print relevant messages to indicate the status of gates and waiting lines during the simulation.
   - Verify that the system handles various scenarios such as empty and full stacks/queues appropriately.

(2 marks)