Question 1

Scenario: A theme park has a ticketing counter where customers arrive one by one. Each customer is given a unique number in the order they arrive. However, due to complaints of long wait times, the park management decides to allow premium customers to be served first, even if they arrive later than regular customers.

To solve this problem, you need to simulate the process of serving customers based on the following rules:

1. Premium customers are served in the order they arrive (*First-In-First-Out for premiums*).
2. Regular customers are served after all premium customers, again in the order they arrive.

Task:

a. Suggest and justify whether a **stack** or a **queue** is better suited for implementing this simulation.

b. Implement the simulation using your chosen data structure(s).

—

Question 2

Scenario: You are designing a text editor's undo and redo system. The editor keeps track of user actions (e.g., typing a word or deleting a line) so that users can undo or redo their changes. The rules are:

1. Undo reverses the most recent action (*Last-In-First-Out*).
2. Redo restores the most recently undone action (if any).

Task:

a. Suggest and justify whether a **stack** or a **queue** is better suited for implementing the undo/redo feature.

b. Implement the undo/redo system using your chosen data structure(s).

—

Question 3

Scenario: You are designing a system for managing deliveries for a logistics company. Deliveries have a priority based on their *urgency level* and *distance*:

1. **Urgency levels** are assigned as follows:
   i. High: Must be delivered immediately.
   ii. Medium: Should be delivered soon but can wait if high-priority deliveries exist.
   iii. Low: Can be delivered after all high and medium-priority deliveries are done.
2. For deliveries with the **same urgency level**, the one with the **shortest distance** should be delivered first.

During peak hours, there is a sudden rule change:

- Deliveries that are **close** (distance $\leq$ 5 km) are prioritized, regardless of their urgency level.
- Deliveries farther than 5 km still follow the original rules.

Task:

a. Suggest and justify whether a **priority queue** or another data structure would be most suitable for implementing this system.

b. Implement the system using your suggested data structure, including the sudden rule change.

—

Question 4

Scenario: You are tasked with designing a software module for a robot that performs calculations based on instructions given in three different notations: **infix**, **postfix**, and **prefix**. Each notation represents the same type of mathematical expression, but the robot's performance varies based on the notation used:

1. **Infix Notation:** Example: $3 + (4 \times 5)$. This is the most human-readable format, but the robot finds it difficult to evaluate directly without converting it into a different form.

2. **Postfix Notation (Reverse Polish Notation):** Example: 3 4 5 $\times$ +. The robot can evaluate this efficiently using a stack because operations appear after operands.

3. **Prefix Notation (Polish Notation):** Example: + 3 $\times$ 4 5. The robot can also evaluate this using a stack, but the process requires a different strategy compared to postfix.

The robot receives instructions in **infix notation** but can evaluate them faster if they are converted to **postfix** or **prefix**.

Task:

a. Convert the given infix expression to both postfix and prefix notations.

b. Use a stack-based approach to evaluate the postfix and prefix expressions.

Given Expression: $3 + (4 \times 5) - 2$