**Question 1: Linked Lists**

**Scenario:** You are tasked with implementing a database system that manages employee records for a company. The system needs to support dynamic insertion, deletion, and retrieval of employee information, where each employee record contains an ID, name, and department.

**Questions:**

1. How would you design a singly linked list to manage the employee records efficiently? Describe the structure of the nodes and the operations (insertion, deletion, search) in detail.
2. How would you extend this design to support a doubly linked list? What advantages would this provide, and how would the operations differ?

**Question 2: Stack (Infix, Prefix, Postfix)**

**Scenario:** You are developing a calculator application that needs to evaluate mathematical expressions provided by the user. The expressions may be in infix notation (e.g., 3 + 5 \* (2 - 8)).

**Questions:**

1. Explain how you would use a stack to convert an infix expression to postfix notation. Provide a step-by-step example of converting the expression 3 + 5 \* (2 - 8) to postfix.
2. Describe how the postfix expression can be evaluated using a stack. Show the evaluation process for the previously converted postfix expression.

**Question 3: Stack (Expression Evaluation)**

**Scenario:** You are designing a system that requires balanced expressions for validation purposes. The system needs to check if expressions containing different types of parentheses are balanced. For example, the expression {[()()]} should be balanced, while {[(])} should not.

**Questions:**

1. How would you implement an algorithm using stacks to check if an expression with multiple types of parentheses is balanced? Describe the algorithm and its steps.
2. What considerations should be taken into account to handle cases where the expression contains nested parentheses or additional characters?

**Question 4: Queue (Priority Queue)**

**Scenario:** You are implementing a task scheduling system where tasks have different priorities, and tasks with higher priorities should be executed before tasks with lower priorities.

**Questions:**

1. How would you design a priority queue to manage and execute tasks based on their priority levels? Explain the data structure you would use and how you would implement it.
2. Provide a detailed explanation of how tasks would be enqueued and dequeued, and describe how the priority queue handles tasks with the same priority.

**Question 5: Linked Lists (Circular Linked List)**

**Scenario:** You are developing a circular linked list to represent a round-robin scheduling system for processes. In this system, each process gets a fixed time slice for execution, and the scheduler cycles through the processes in a circular manner.

**Questions:**

1. Describe the design and implementation of a circular linked list for this scheduling system. Include details on the node structure and the operations (insertion, deletion, traversal).
2. How does the circular nature of the linked list benefit the round-robin scheduling algorithm, and what are the potential challenges or limitations of using a circular linked list for this purpose?