**Linked List**

**📘 Theory Questions (Basic to Intermediate)**

**🔹 Basic**

1. What is a linked list?
2. What are the types of linked lists?
3. What is the difference between singly and doubly linked lists?
4. What are the advantages of a linked list over arrays?
5. What are the disadvantages of linked lists?
6. How is memory managed in a linked list?
7. Why is random access not possible in a linked list?
8. What is the time complexity of insertion/deletion in a linked list?

**🔹 Intermediate**

1. What is a circular linked list?
2. How is a circular linked list different from a regular linked list?
3. What is a sentinel node or dummy node?
4. When would you prefer a doubly linked list over a singly linked list?
5. How do you detect a loop in a linked list?
6. How do you remove a loop in a linked list?

**💻 Coding Questions (With Increasing Difficulty)**

**🔸 Basic**

1. **Insert a node at the beginning / end / any position**
2. **Delete a node from beginning / end / specific position**
3. **Find length of a linked list**
4. **Search an element in the linked list**
5. **Reverse a linked list**

**🔸 Intermediate**

1. **Find the middle of a linked list**
2. **Find the Nth node from the end**
3. **Detect and remove loop (Floyd’s Cycle Detection)**
4. **Check if a linked list is a palindrome**
5. **Merge two sorted linked lists**
6. **Remove duplicates from a sorted linked list**
7. **Remove duplicates from an unsorted linked list**

**🔸 Advanced**

1. **Intersection point of two linked lists**
2. **Add two numbers represented by linked lists**
3. **Flatten a multilevel linked list**
4. **Clone a linked list with random pointers**
5. **Sort a linked list using merge sort**
6. **Segregate even and odd nodes**
7. **Delete N nodes after M nodes**
8. **Rotate a linked list by K places**
9. **LRU Cache implementation using Linked List and HashMap**

**📂 Bonus System Design-Level**

* Implement a **Queue/Stack using Linked List**
* Design a **Browser history** system using Doubly Linked List
* Implement a **Music Playlist** using a Circular Linked List