

# DSA Practice Questions Assignment

## **1. Define a doubly linked list**

A doubly linked list is a data structure where each node contains a value and two pointers: one pointing to the next node and another pointing to the previous node.

## **2. Write a function to reverse a linked list in-place**

Reverse a linked list by iterating through it and swapping the next and previous pointers of each node until the list is completely reversed.

## **3. Detect cycle in a linked list**

Use Floyd's Cycle-Finding Algorithm (Tortoise and Hare), where two pointers traverse the list at different speeds; if they meet, a cycle exists.

## **4. Merge two sorted linked lists into one**

Traverse both lists, compare the nodes, and merge them into a new sorted linked list by connecting the smaller node first.

## **5. Write a function to remove nth node from the end in a linked list**

Use two pointers: move one pointer n steps ahead, then move both pointers one step at a time until the first pointer reaches the end. The second pointer will be at the nth node from the end.

## **6. Remove duplicates from a sorted linked list**

Traverse the list and compare each node with the next one. If they are the same, remove the next node.

## **7. Find the intersection of the two linked lists**

Traverse both lists to find the first common node. If there's no common node, return null.

## **8. Rotate a linked list by k positions to the right**

Find the length of the list, calculate the effective number of rotations ( $k \% \text{length}$ ), and move the head of the list accordingly.

## **9. Add Two Numbers Represented by Linked Lists**

Traverse both lists, add the corresponding digits along with a carry, and create a new list to store the sum.

#### **10. Clone a Linked List with next and Random Pointer**

**Create a copy of each node, insert it next to the original node, and then set the arbit pointer for each copied node based on the original node's arbit pointer.**