



# SABUDH

## 1. Print the Middle of a given linked list

**Given a singly linked list, find the middle of the linked list. For example, if the given linked list is 1->2->3->4->5 then the output should be 3.**

**If there are even nodes, then there would be two middle nodes, we need to print the second middle element.**

For example, if the given linked list is 1->2->3->4->5->6 then the output

should be 4. Test Cases:

TestCase 1:

2->3->4->5->NULL

The middle element is 4

TestCase 2:

1->2->3->4->5->NULL

The middle element is 3

## 2. Delete Middle of Linked List

**Given a singly linked list, delete the middle of the linked list. For example, if given linked list is 1->2->3->4->5->NULL, then the linked list should be modified to 1->2->4->5->NULL.**

**If there are even nodes, then there would be two middle nodes, we need to delete the second middle element.**

For example, if given linked list is 1->2->3->4->5->6->NULL then it should be modified to 1->2->3->5->6->NULL.

If the input linked list is NULL or has 1 node, then it should return

NULL Example 1:



Input:

LinkedList: 1->2->3->4->5->NULL

Output: 1 2 4 5

Example 2:

Input:

LinkedList: 2->4->6->7->5->1->NULL

Output: 2 4 6 5 1

### 3. Remove duplicate elements from sorted linked list

Write a function that takes a list sorted in non-decreasing order and deletes any duplicate nodes from the list. The list should only be traversed once.

For example if the linked list is 11->11->11->21->43->43->60->NULL then removeDuplicates() should convert the list to 11->21->43->60->NULL.

Test Cases:

TestCase1:

Linked list: 11->11->11->13->13->20->NULL

Output: 11->13->20->NULL

TestCase2:

Linked list: 10->15->15->15->20->20->20->23->25->25->NULL

Output: 10->15->20->23->25->NULL



#### 4. Reverse a Linked List

**Given a pointer to the head node of a linked list, the task is to reverse the linked list. We need to reverse the list by changing the links between nodes.**

Examples:

Input: Head of following linked list

1->2->3->4->NULL

Output: Linked list should be changed to,

4->3->2->1->NULL

Input: Head of following linked list

1->2->3->4->5->NULL

Output: Linked list should be changed to,

5->4->3->2->1->NULL

#### 5. Add 1 to a number which is represented as a linked list

**A number is represented as a linked list such that each digit corresponds to a node in the linked list. Add 1 to the number and form a new linked list. For example 1999 is represented as (1-> 9-> 9 -> 9->NULL) and adding 1 to it should change it to (2->0->0->0->NULL)**

TestCase 1:

Input: 1999

Output: 2000

TestCase 2:

Input: 3453

Output: 3454



## 6. Add two numbers represented by Linked List

**Given two numbers represented by two lists, write a function that returns the sum in the form of a linked list.**

Example:

Input:

List1: 5->6->3->NULL // represents number 563

List2: 8->4->2->NULL // represents number 842

Output:

Resultant list: 1->4->0->5->NULL // represents number 1405

Explanation:  $563 + 842 = 1405$

Input:

List1: 7->5->9->4->6->NULL // represents number 75946

List2: 8->4->NULL // represents number 84

Output:

Resultant list: 7->6->0->3->0->NULL // represents number 76030

Explanation:  $75946+84=76030$

## 7. Find the second last element from the linked list

Input : List = 2 -> 4 -> 6 -> 8 -> 33 -> 67 ->

NULL Output : 33

Input : List = 1 -> 2 -> 3 -> 4 -> 5 ->

NULL Output : 4