



# Aror University of Art, Architecture, Design & Heritage Sukkur

Department of AI-Multimedia and Gaming

---

Lab 01: Fundamentals of Linked List data structure Date: Sep 10, 2024 Subject: Data

Structure (CSC221), Fall 2024 Instructor: Abdul Ghafoor

**Lab objectives:** Objective: To practice and understand the basic operations in a singly linked list, including insertion, deletion, Searching, and traversal of nodes.

**NAME:** MUHAMMAD SAEED SHAIKH

**ROLL NO:** 0069

## Lab Part\_01

### Github link for task

[https://github.com/Saeed-Shaikh01/DSA\\_LAB\\_02\\_Task-1](https://github.com/Saeed-Shaikh01/DSA_LAB_02_Task-1)

## Lab Part 02: LeetCode

### Task 01:

<https://leetcode.com/problems/intersection-of-two-linked-lists/description/?envType=problem-list-v2&envId=linked-list>

**160. Intersection of Two Linked Lists** Solved ✓

Easy Topics Companies

Given the heads of two singly linked-lists `headA` and `headB`, return the node at which the two lists intersect. If the two linked lists have no intersection at all, return `null`.

For example, the following two linked lists begin to intersect at node `c1`:

A:

B:

The test cases are generated such that there are no cycles anywhere in the entire linked structure.

**Note** that the linked lists must **retain their original structure** after the function returns.

15.1K 136

```
public ListNode getIntersectionNode(ListNode headA, ListNode headB) {
    if(headA == null || headB == null)
        return null;

    ListNode a = headA;
    ListNode b = headB;

    while( a != b){
        a = a == null? headB : a.next;
        b = b == null? headA : b.next;
    }
}
```

Testcase | Test Result

Case 1 Case 2 Case 3 +

intersectVal =

8

Source ?

Task 02:

<https://leetcode.com/problems/remove-duplicates-from-sorted-list/description/?envType=problem-list-v2&envId=linked-list>

```
11 class Solution {
12     public ListNode deleteDuplicates(ListNode head) {
13
14         if (head == null) {
15             return head;
16         }
17
18         ListNode current = head;
19         while (current.next != null) {
20             if (current.val == current.next.val) {
21                 current.next = current.next.next;
22             } else {
23                 current = current.next;
24             }
25         }
26         return head;
27     }
```

Saved

Ln 13, Col 1

☒ Testcase | [> Test Result](#)

Output

[1,2]

Task 03:

<https://leetcode.com/problems/merge-two-sorted-lists/description/?envType=problem-list-v2&envId=linked-list>

```
Code
Java Auto
public ListNode mergeTwoLists(ListNode list1, ListNode list2) {
    if(list1!=null && list2!=null){
        if(list1.val<list2.val){
            list1.next=mergeTwoLists(list1.next,list2);
            return list1;
        }
        else{
            list2.next=mergeTwoLists(list1,list2.next);
            return list2;
        }
    }
    if(list1==null)
        return list2;
    return list1;
}
```

Saved Ln 22, Col 1

☒ Testcase | >\_ Test Result

Output

[1,1,2,3,4,4]

Task 04:

<https://leetcode.com/problems/add-two-numbers/description/?envType=problem-list-v2&envId=linked-list>

```
</> Code ⌵
Java ⌵ 🔒 Auto ☰ 🔖 { } ↶

6      ListNode list1 = new ListNode();
7      ListNode list = list1;
8
9      while (l1 != null || l2 != null) {
10         int x = (l1 != null) ? l1.val : 0;
11         int y = (l2 != null) ? l2.val : 0;
12         int sum = x + y + carry;
13
14         list.next = new ListNode(sum % 10);
15         carry = sum / 10;
16         list = list.next;
17
18         if (l1 != null) l1 = l1.next;
19         if (l2 != null) l2 = l2.next;
20     }
21
22     if (carry > 0) {
```