WORKSHEET 2

Student Name: Hardik Jhamb **UID:** 22BCS16504

Branch: BE-CSE **Section/Group:** 22BCS_NTPP-602-A

Semester: 6th **Date of Performance:**

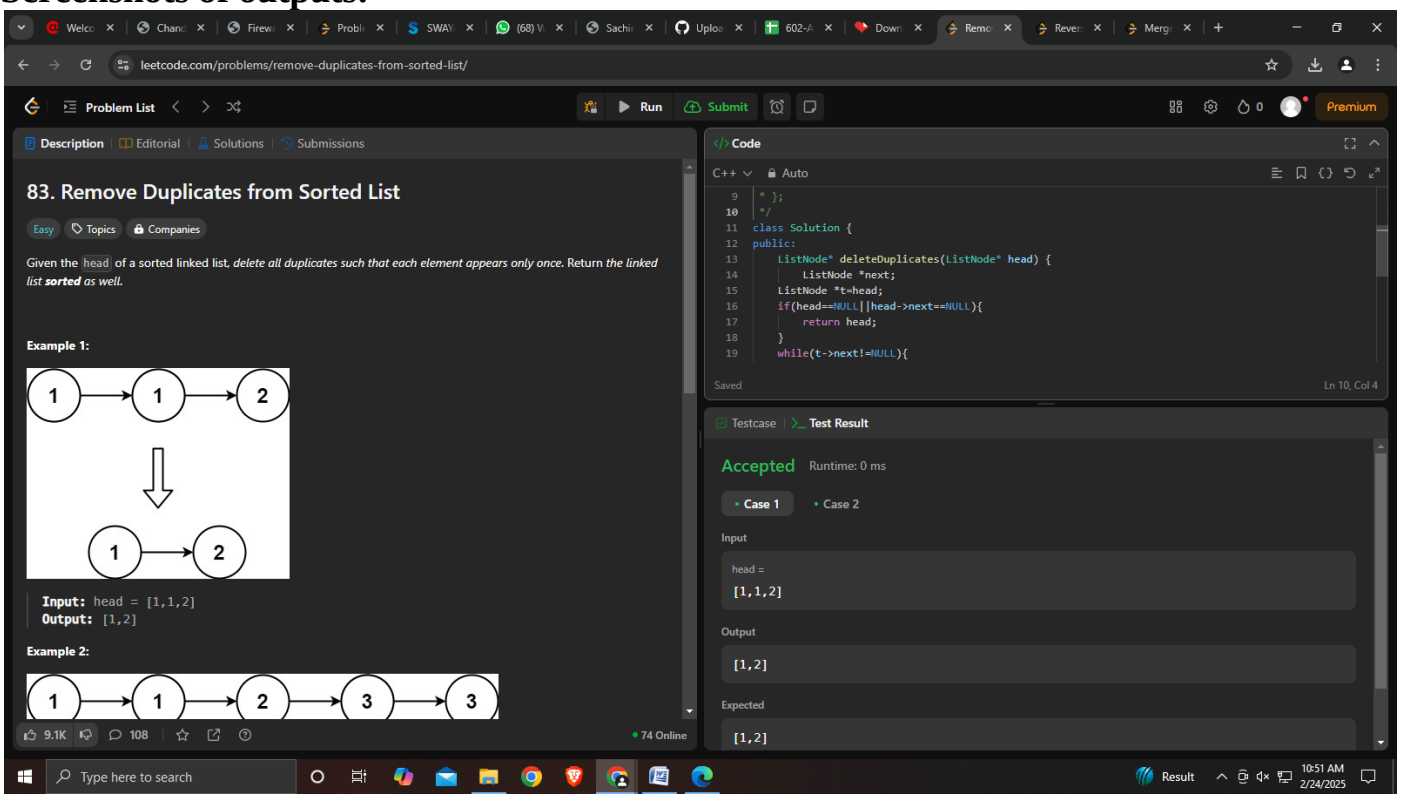
Subject Name: AP LAB - II **Subject Code:** 22CSP-351

1. **Aim:** Remove duplicates from a sorted list
   Reverse a linked list
   Merge two sorted linked lists

2. **Source Code:**

```
/**
 * Definition for singly-linked list.
 * struct ListNode {
 *     int val;
 *     ListNode *next;
 *     ListNode() : val(0), next(nullptr) {}
 *     ListNode(int x) : val(x), next(nullptr) {}
 *     ListNode(int x, ListNode *next) : val(x), next(next) {}
 * };
 */
class Solution {
public:
    ListNode* deleteDuplicates(ListNode* head) {
        ListNode *next;
    ListNode *t=head;
    if(head==NULL||head->next==NULL){
```

3. Screenshots of outputs:



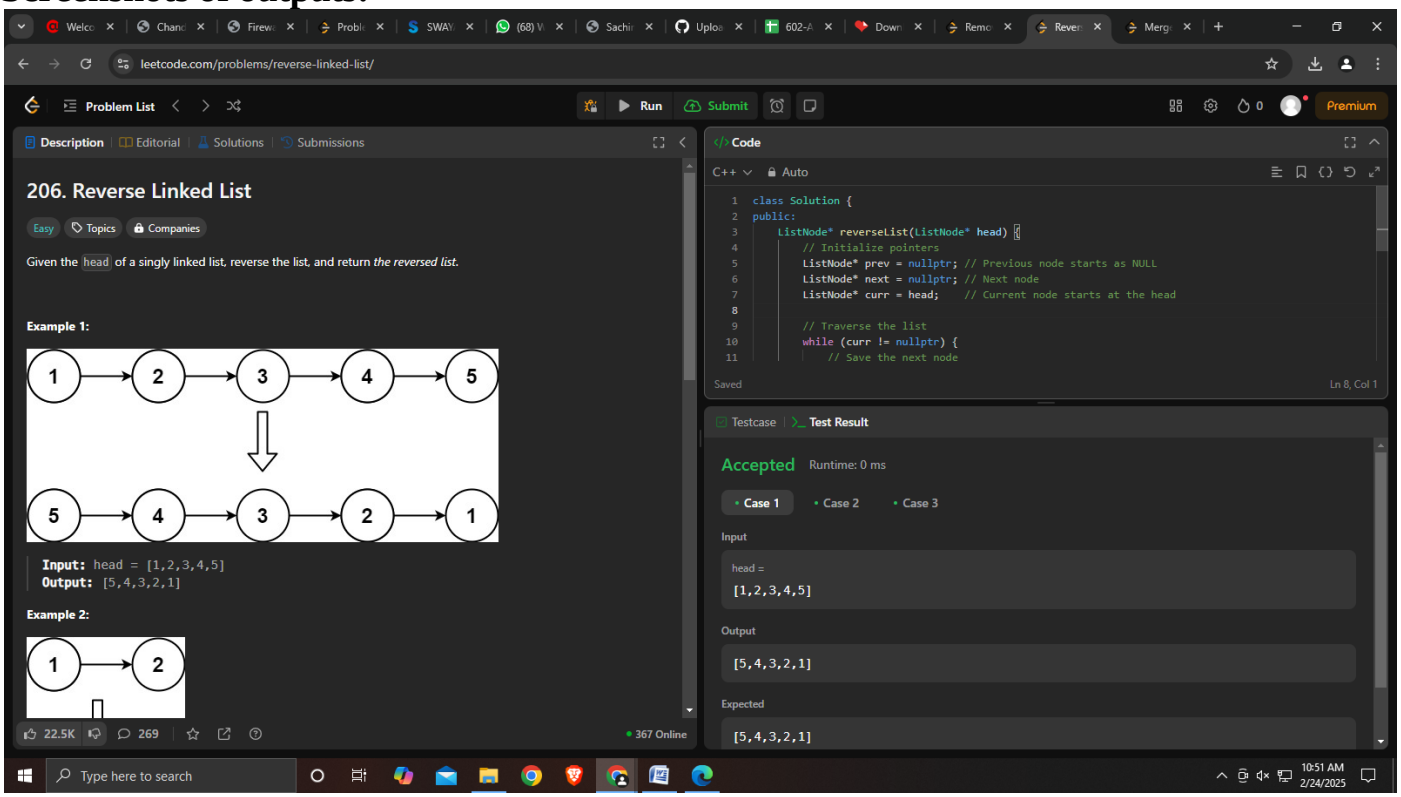
2. Aim: Given the head of a singly linked list, reverse the list, and return *the reversed list*.

Source Code:

```
class Solution {
public:
    ListNode* reverseList(ListNode* head) {
        // Initialize pointers
        ListNode* prev = nullptr; // Previous node starts as NULL
        ListNode* next = nullptr; // Next node
        ListNode* curr = head;    // Current node starts at the head

        // Traverse the list
        while (curr != nullptr) {
            // Save the next node
            next = curr->next;
```

Screenshots of outputs:



Aim: You are given the heads of two sorted linked lists list1 and list2.

Merge the two lists into one **sorted** list. The list should be made by splicing together the nodes of the first two lists.

Return *the head of the merged linked list*.

Source Code:

```
/**
 * Definition for singly-linked list.
 * struct ListNode {
 *     int val;
 *     ListNode *next;
 *     ListNode() : val(0), next(nullptr) {}
 *     ListNode(int x) : val(x), next(nullptr) {}
 *     ListNode(int x, ListNode *next) : val(x), next(next) {}
 * };
 */
class Solution {
public:
    ListNode* mergeTwoLists(ListNode* list1, ListNode* list2) {
        ListNode* Dummy = new ListNode(0); // farzi Node
```

4. Screenshots of outputs: