Assignment 3

Advanced Programming Lab – II

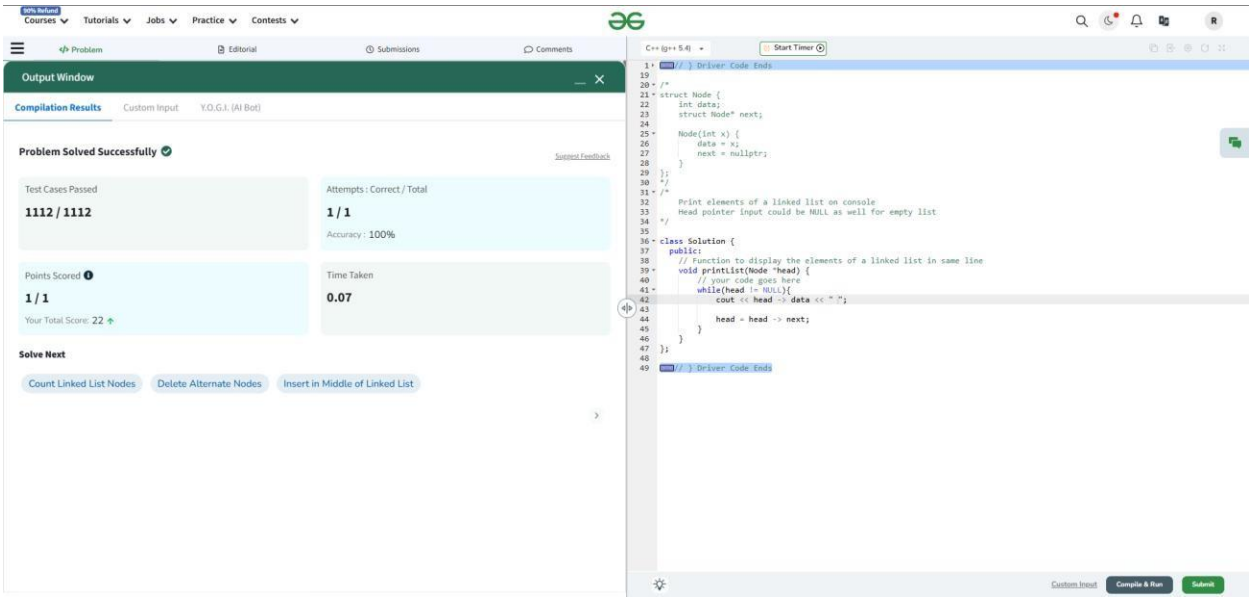
Submitted By – Lakshyaraj Singh

UID – 22BCS10687

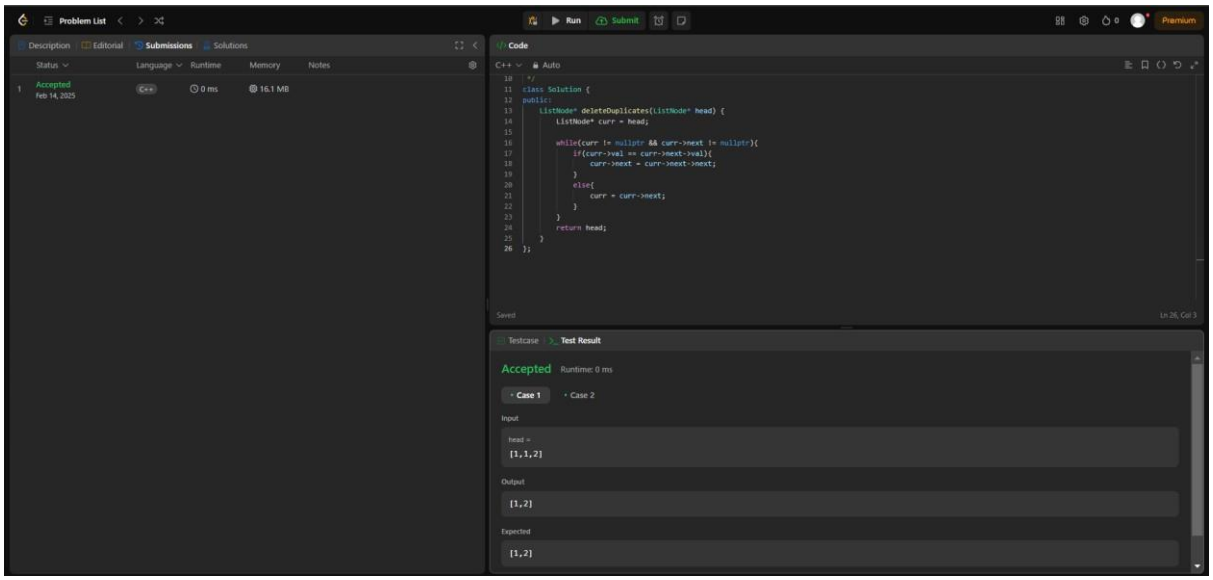
Section – 22BCS_IOT – 610

Submitted To – Pratima Sonali Horo (E18304)

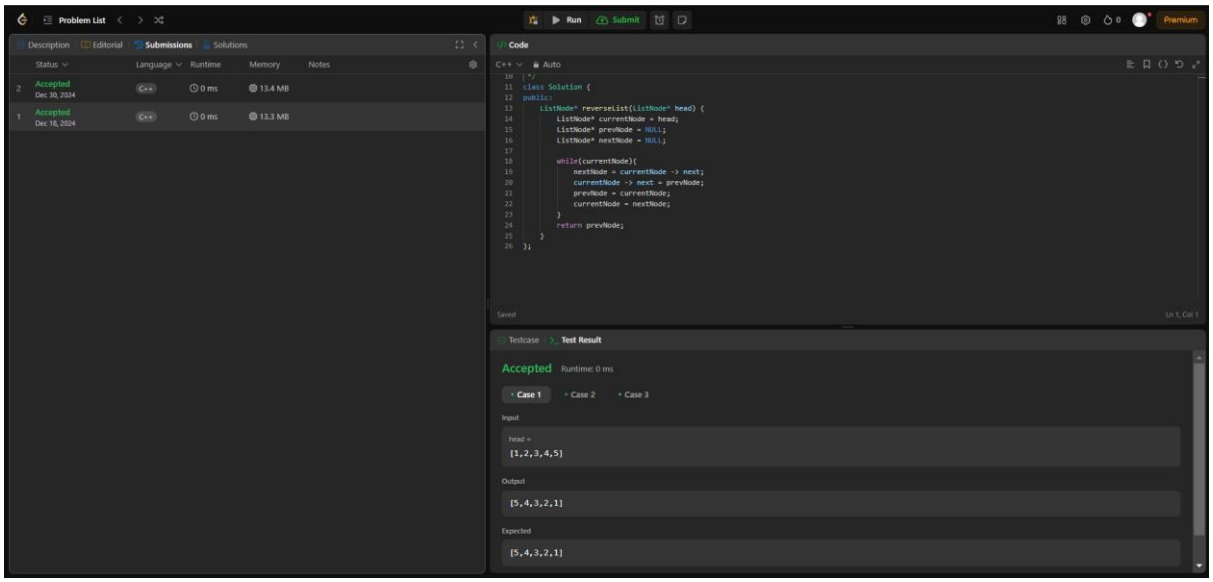
1. Print Linked List: https://www.geeksforgeeks.org/problems/print-linked-list-elements/0



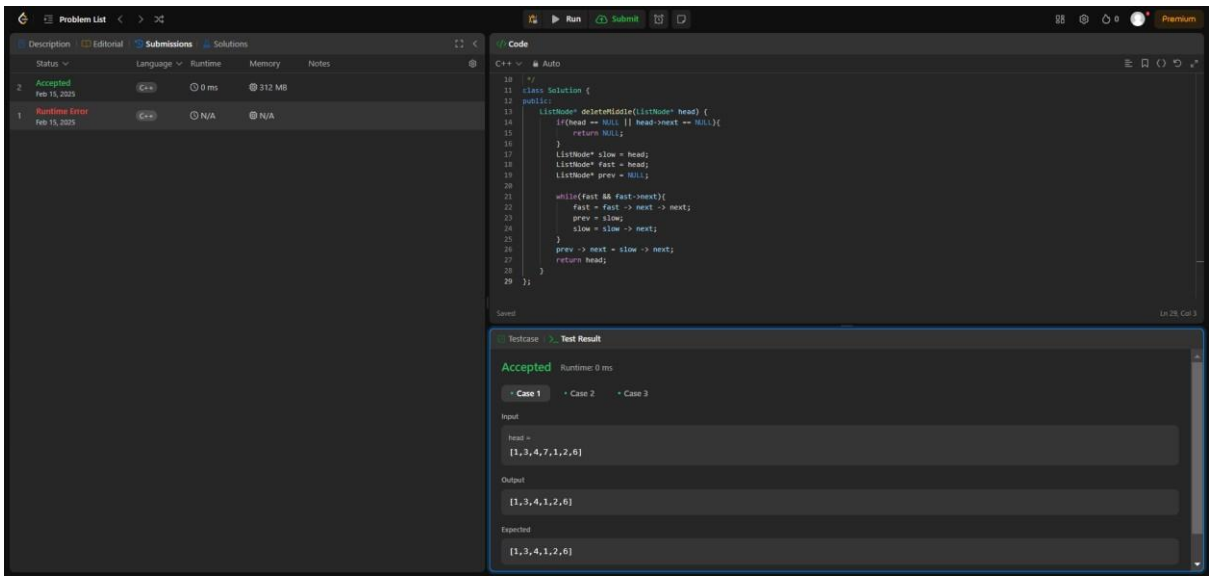
2. Remove Duplicates from a Sorted List: https://leetcode.com/problems/remove-duplicates-from-sorted-list/description/



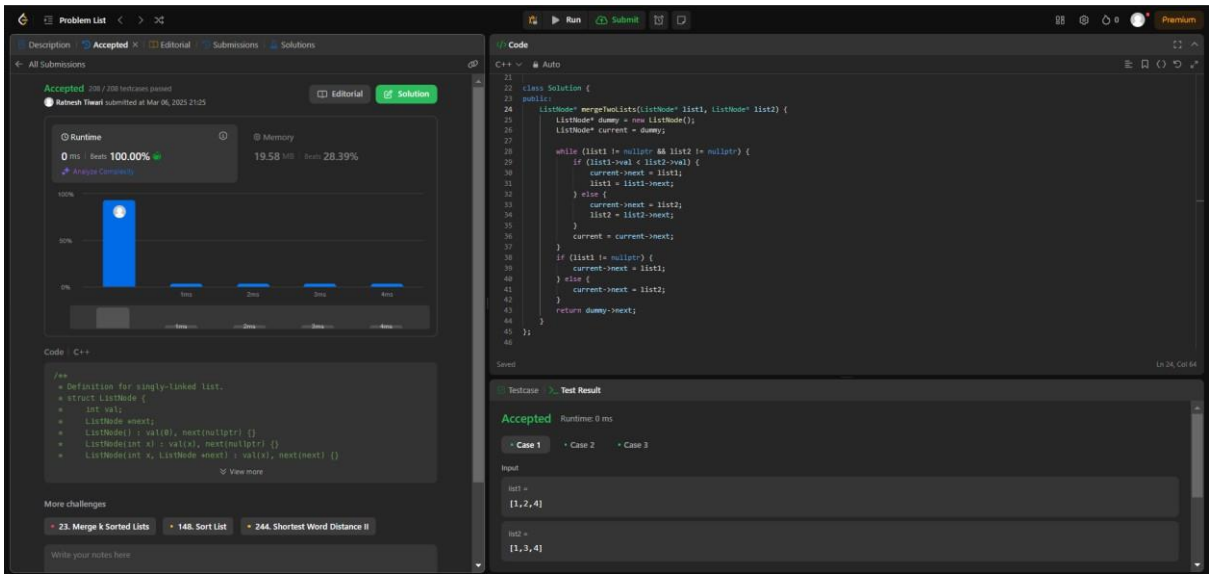
3. Reverse a Linked List: https://leetcode.com/problems/reverse-linked-list/description/



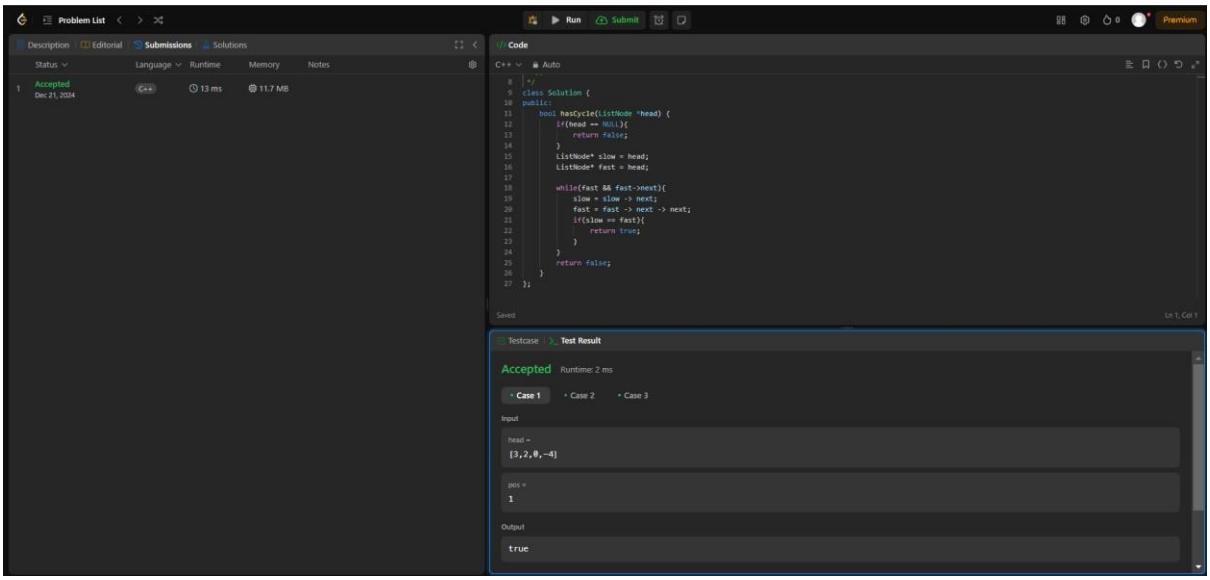
4. Delete middle node of a list: https://leetcode.com/problems/delete-the-middle-node-of-a-linked-list/description/



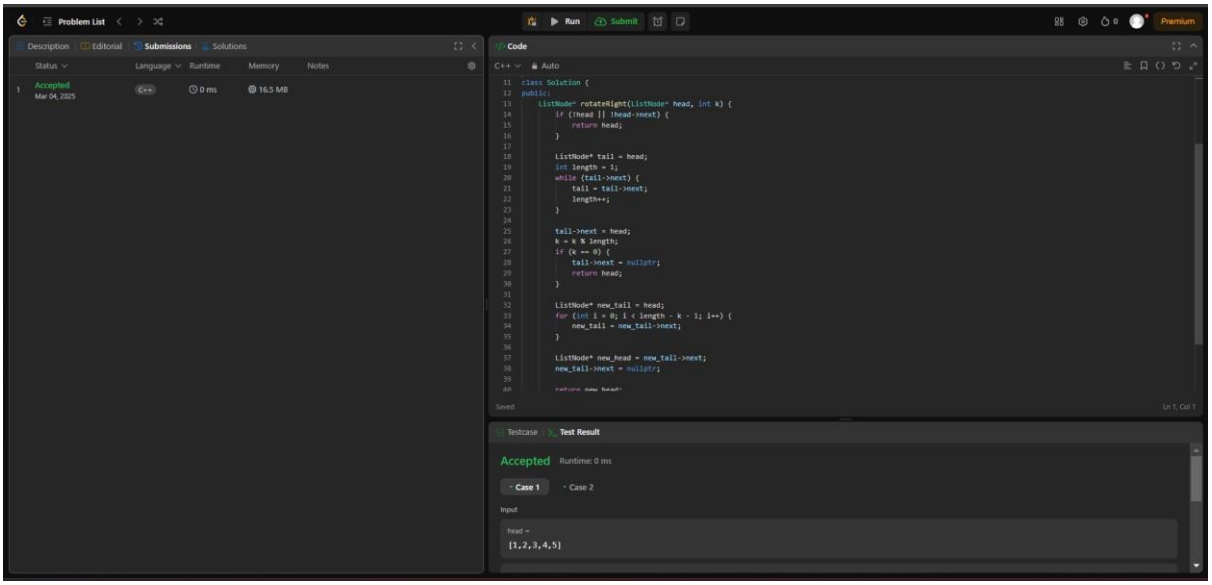
5. Merge two sorted linked lists: https://leetcode.com/problems/merge-two-sorted-lists/description/



6. Detect a cycle in a linked list: https://leetcode.com/problems/linked-list-cycle/description/



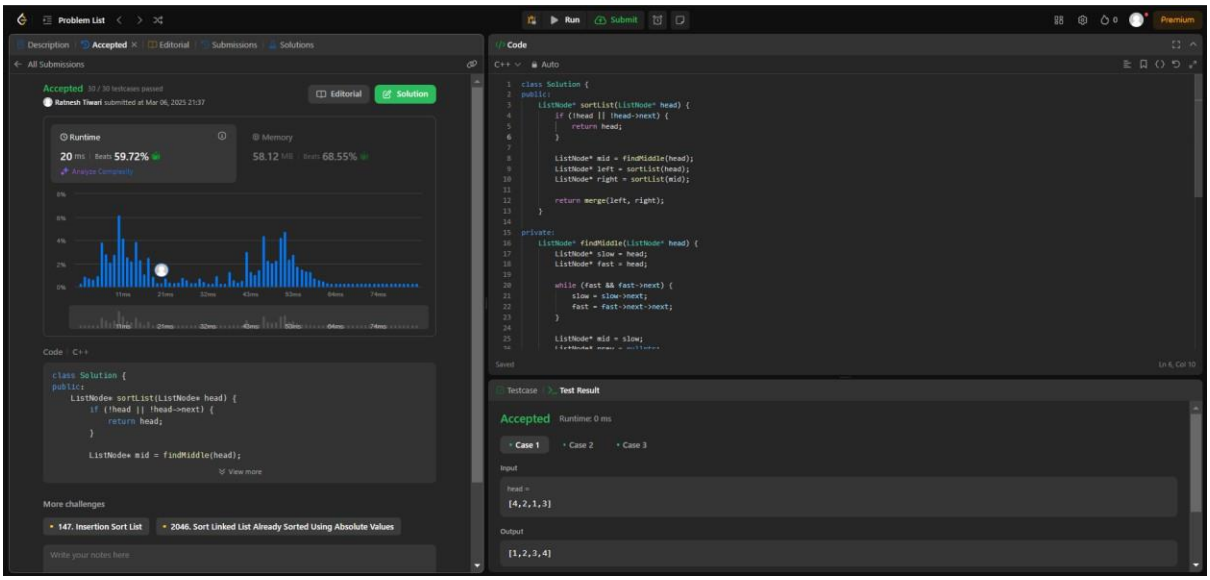
7. Rotate a list: https://leetcode.com/problems/rotate-list/description/



8. Sort List: https://leetcode.com/problems/sort-list/description/

```
class Solution {
public:
    ListNode* sortList(ListNode* head) {
        if (!head || !head->next) {
            return head;
        }

        ListNode* mid = findMiddle(head);
        ListNode* left = sortList(head);
        ListNode* right = sortList(mid);

        return merge(left, right);
    }

private:
    ListNode* findMiddle(ListNode* head) {
        ListNode* slow = head;
        ListNode* fast = head;

        while (fast && fast->next) {
            slow = slow->next;
            fast = fast->next->next;
        }

        ListNode* mid = slow;
        ListNode* prev = nullptr;

        while (head != mid) {
            prev = head;
            head = head->next;
        }

        if (prev) {
            prev->next = nullptr;
        }

        return mid;
    }

    ListNode* merge(ListNode* left, ListNode* right) {
        if (!left) return right;
        if (!right) return left;
```

```
        if (left->val < right->val) {
            left->next = merge(left->next, right);
            return left;
        } else {
            right->next = merge(left, right->next);
            return right;
        }
    }
};
```



9. Merge k sorted lists: https://leetcode.com/problems/merge-k-sorted-lists/description/