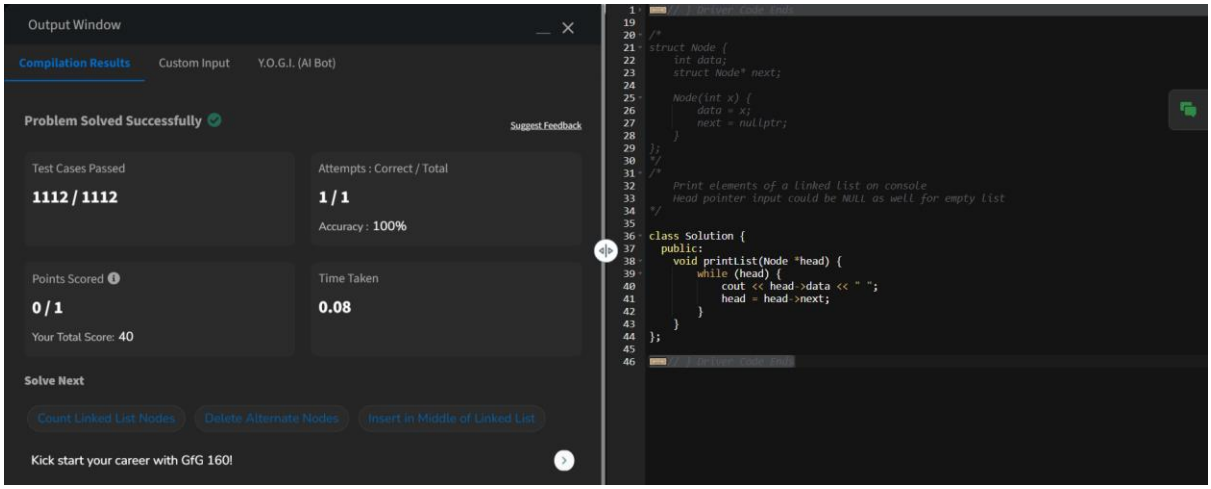
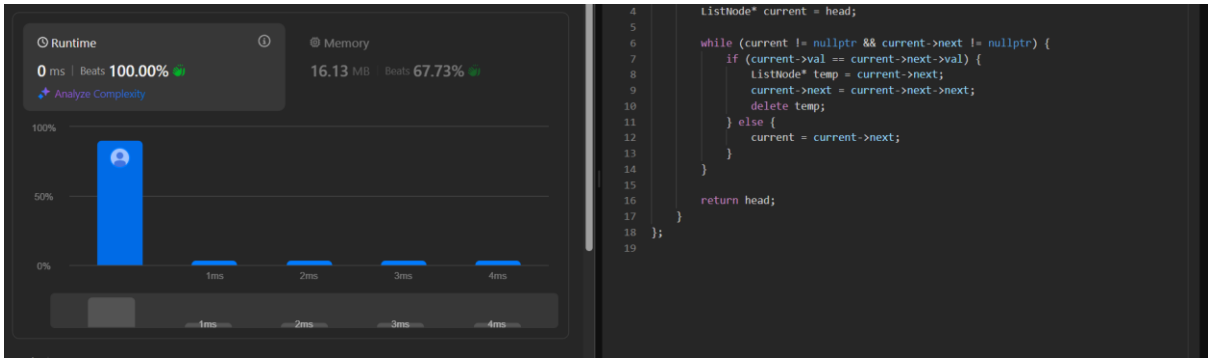
NAME: SHATAKSHI SHARMA, UID: 22BCS10768

AP – ASSIGNMENT 3

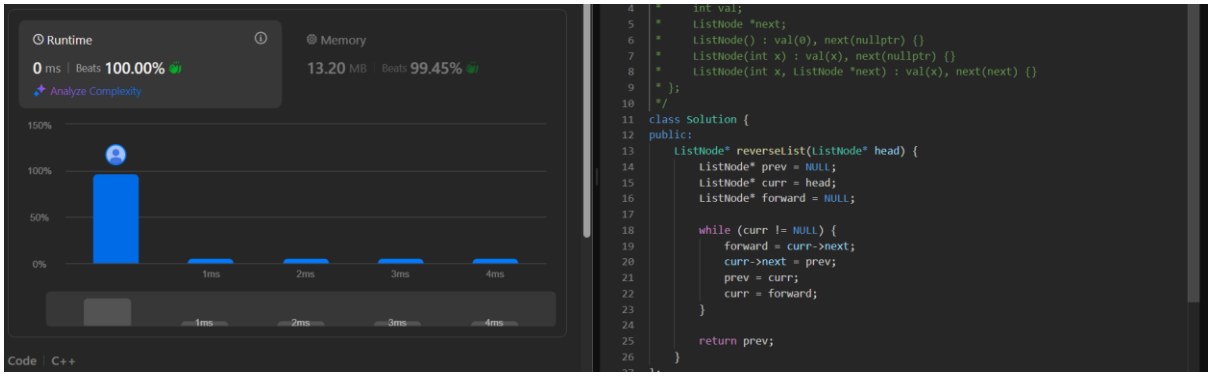
Q1. Print Linked List:



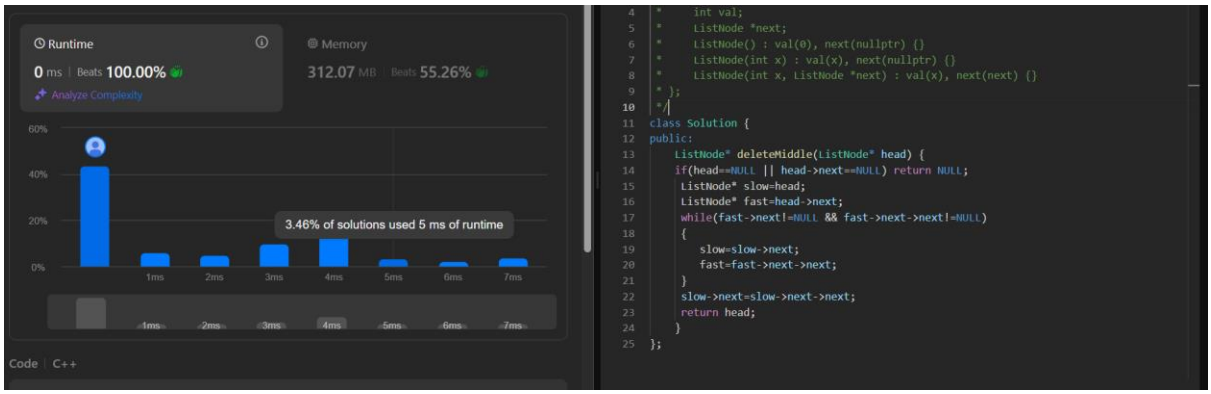
Q2. Remove duplicates from a sorted list:



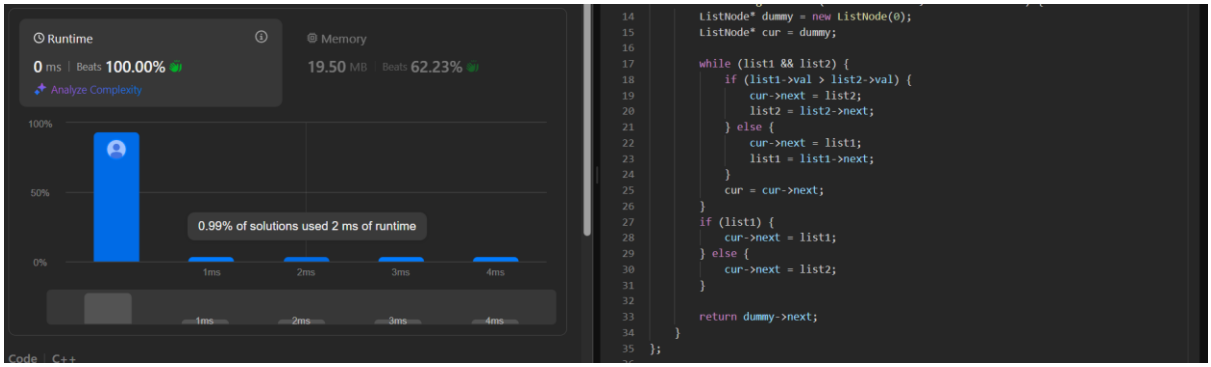
Q3. Reverse a linked list:



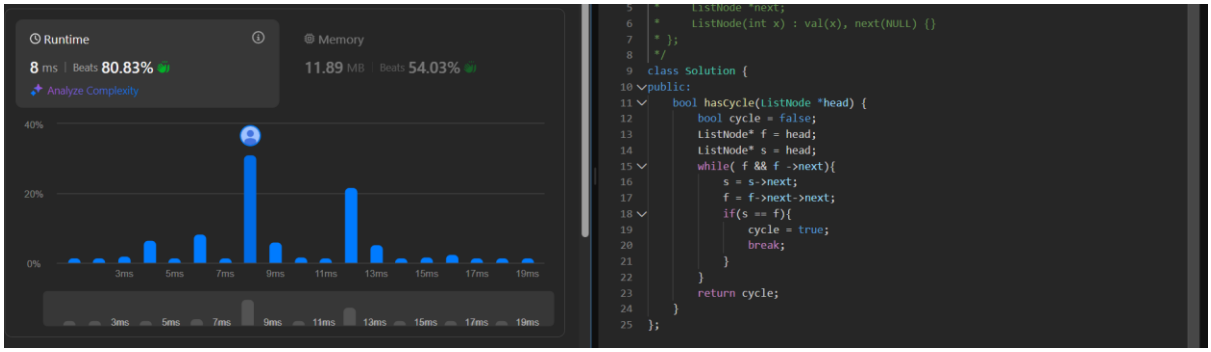
Q4. Delete middle node of a list:



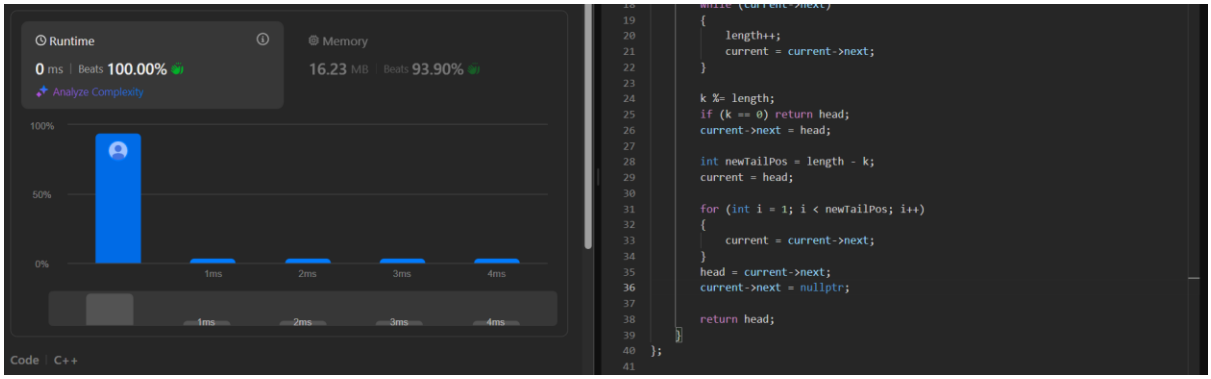
Q5. Merge two sorted linked lists:



Q6. Detect a cycle in a linked list:

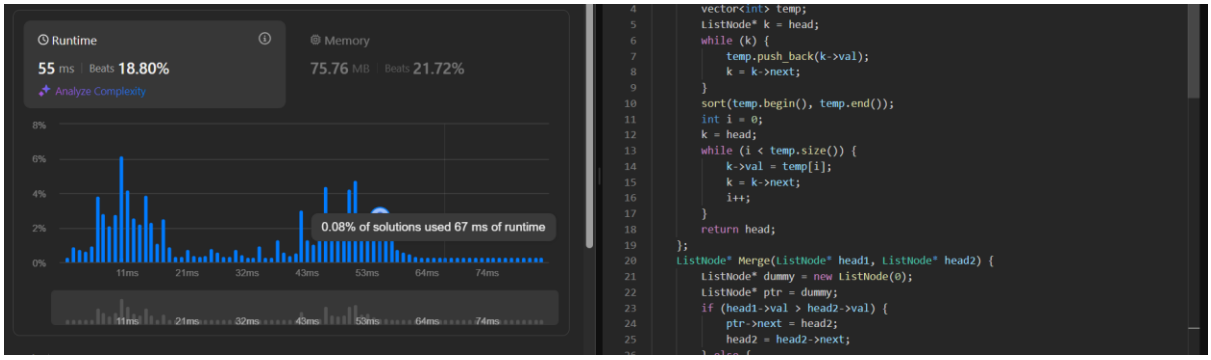


Q7. Rotate a list:



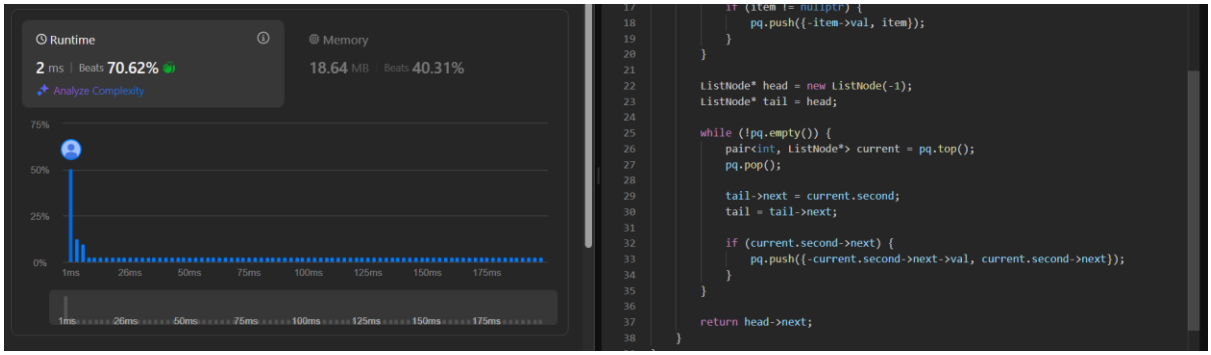

```
        {
            length++;
            current = current->next;
        }

        k %= length;
        if (k == 0) return head;
        current->next = head;

        int newTailPos = length - k;
        current = head;

        for (int i = 1; i < newTailPos; i++)
        {
            current = current->next;
        }
        head = current->next;
        current->next = nullptr;

        return head;
    }
};
```

Q8. Sort List:




```
        vector<int> temp;
        ListNode* k = head;
        while (k) {
            temp.push_back(k->val);
            k = k->next;
        }
        sort(temp.begin(), temp.end());
        int i = 0;
        k = head;
        while (i < temp.size()) {
            k->val = temp[i];
            k = k->next;
            i++;
        }
        return head;
    }
};
ListNode* Merge(ListNode* head1, ListNode* head2) {
    ListNode* dummy = new ListNode(0);
    ListNode* ptr = dummy;
    if (head1->val > head2->val) {
        ptr->next = head2;
        head2 = head2->next;
```

Q9. Merge k sorted lists:




```
            pq.push({-item->val, item});
        }
    }

    ListNode* head = new ListNode(-1);
    ListNode* tail = head;

    while (!pq.empty()) {
        pair<int, ListNode*> current = pq.top();
        pq.pop();

        tail->next = current.second;
        tail = tail->next;

        if (current.second->next) {
            pq.push({-current.second->next->val, current.second->next});
        }
    }

    return head->next;
}
```