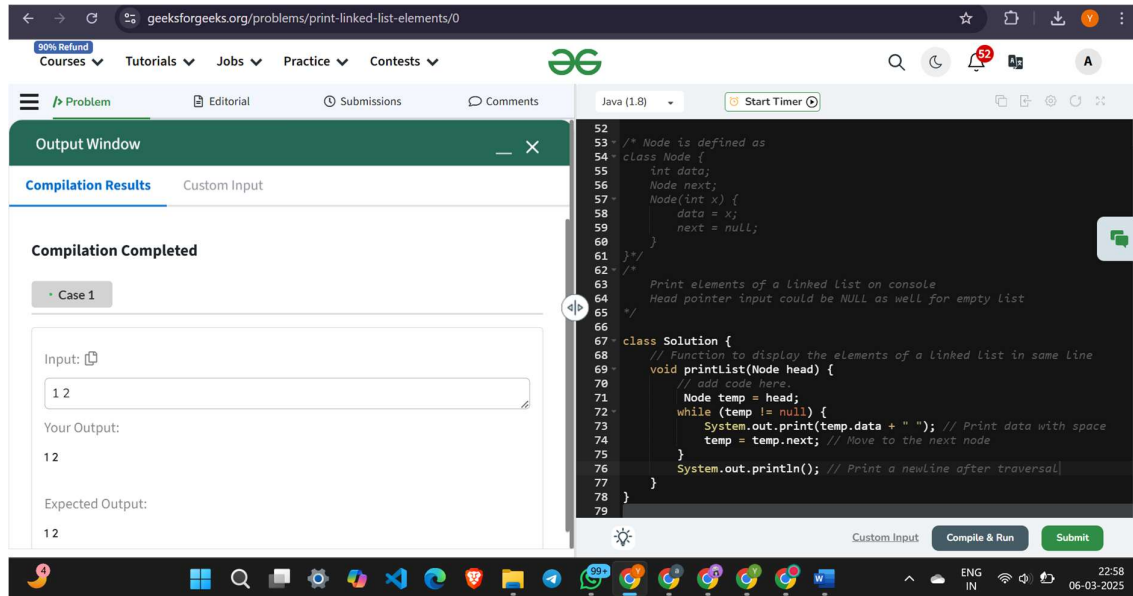
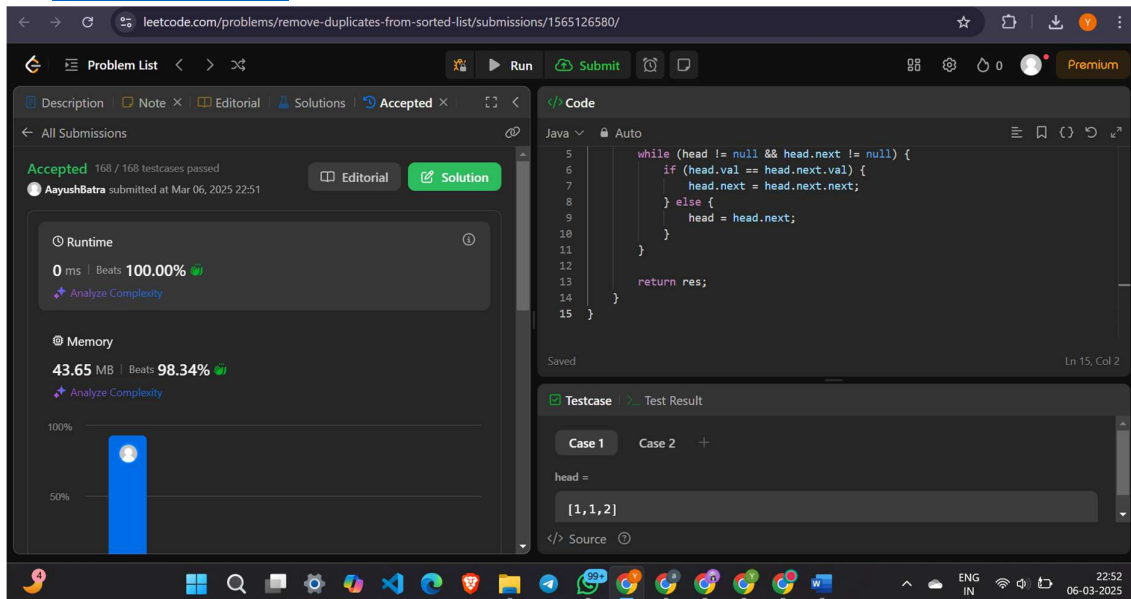


AP Assignment 3

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/>

The screenshot shows the LeetCode interface for the 'Reverse a linked list' problem. The submission is accepted, with 28/28 test cases passed. The runtime is 0 ms, beating 100.00% of submissions. The memory usage is 42.72 MB, beating 28.26% of submissions. The code is written in Java and implements a recursive solution to reverse the linked list. The test case shows the input list [1, 2, 3, 4, 5] and the expected output [5, 4, 3, 2, 1].

```
Java  
19 // Move the previous pointer to the current node (since the current node is  
20 prev = curr;  
21  
22 // Move the current pointer to the next node (to continue processing the r  
23 curr = next;  
24 }  
25  
26 // When we've processed all nodes, the prev pointer will be at the new head of  
27 return prev;  
28  
29 }
```

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

The screenshot shows the LeetCode interface for the 'Delete the middle node of a linked list' problem. The submission is accepted, with 70/70 test cases passed. The runtime is 3 ms, beating 99.71% of submissions. The memory usage is 63.14 MB, beating 77.77% of submissions. The code is written in Java and implements a solution to delete the middle node of a linked list. The test case shows the input list [1, 3, 4, 7, 1, 2, 6] and the expected output [1, 3, 4, 7, 2, 6].

```
Java  
9  
10 ListNode dummy = new ListNode(0, head);  
11  
12 temp = dummy;  
13 for (int i = 0; i < count / 2; i++) {  
14     temp = temp.next;  
15 }  
16 temp.next = temp.next.next;  
17 return dummy.next;  
18  
19 }
```

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

Problem List > > > Merge Two Sorted Lists

Accepted 208 / 208 testcases passed
AayushBatra submitted at Mar 06, 2025 22:53

Runtime: 0 ms | Beats 100.00%
Memory: 42.62 MB | Beats 44.78%

```
14     temp = temp.next;
15 }
16
17 if (list1 != null)
18     temp.next = list1;
19 else
20     temp.next = list2;
21
22 return newNode.next;
23
24 }
```

Testcase: Case 1 Case 2 Case 3 +
list1 = [1,2,4]

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

Problem List > > > Linked List Cycle

Accepted 29 / 29 testcases passed
AayushBatra submitted at Mar 06, 2025 22:54

Runtime: 0 ms | Beats 100.00%
Memory: 44.63 MB | Beats 38.30%

```
3     ListNode slow_pointer = head, fast_pointer = head;
4     while (fast_pointer != null && fast_pointer.next != null) {
5         slow_pointer = slow_pointer.next;
6         fast_pointer = fast_pointer.next.next;
7         if (slow_pointer == fast_pointer) {
8             return true;
9         }
10    }
11    return false;
12
13 }
```

Testcase: Case 1 Case 2 Case 3 +
head = [3,2,0,-4]

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

MEGA x Download file x CU-Assignme x Rotate List - L x Sort List - Lee x Merge k Sort x Print Linked L x +

leetcode.com/problems/rotate-list/submissions/1565129722/

Problem List < > x Run Submit x Premium

Description Accepted x Note x Editorial x Solutions x

All Submissions

Accepted 232 / 232 testcases passed
AayushBatra submitted at Mar 06, 2025 22:54

Editorial Solution

Runtime
0 ms | Beats 100.00%
Analyze Complexity

Memory
42.67 MB | Beats 47.83%
Analyze Complexity

Code

```
1 class Solution {
2     public ListNode rotateRight(ListNode head, int k) {
3         if (head == null || head.next == null || k == 0) {
4             return head;
5         }
6
7         int length = 1;
8         ListNode temp = head;
9
10        while (temp.next != null) {
11            temp = temp.next;
12            length++;
13        }
14    }
15 }
```

Testcase Test Result

Case 1 Case 2 +

head =

[1, 2, 3, 4, 5]

Source

22:55 06-03-2025

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

MEGA x Download file x CU-Assignments x Sort List - LeetCode x Merge k Sorted Lists x Print Linked List x +

leetcode.com/problems/sort-list/submissions/1565131462/

Problem List < > x Run Submit x Premium

Description Note x Editorial x Solutions x Accepted x

All Submissions

Accepted 30 / 30 testcases passed
AayushBatra submitted at Mar 06, 2025 22:56

Editorial Solution

Runtime
9 ms | Beats 94.53%
Analyze Complexity

Memory
56.99 MB | Beats 45.43%
Analyze Complexity

Code

```
31         } else {
32             tail.next = l2;
33             l2 = l2.next;
34         }
35         tail = tail.next;
36     }
37
38     tail.next = (l1 != null) ? l1 : l2;
39     return dummy.next;
40 }
41 }
```

Testcase Test Result

Case 1 Case 2 Case 3 +

head =

[4, 2, 1, 3]

Source

22:56 06-03-2025

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

MEGA Download file | jil CU-Assignments/ Sort List - LeetCode Merge k Sorted Lists Print Linked List

leetcode.com/problems/merge-k-sorted-lists/submissions/1565131619/

Problem List Run Submit

Description Note Editorial Solutions Accepted Submit Ctrl Enter

All Submissions

Accepted 134 / 134 testcases passed
AayushBatra submitted at Mar 06, 2025 22:56

Editorial Solution

Runtime
2 ms | Beats 84.11%

Analyze Complexity

Memory
44.40 MB | Beats 78.05%

Analyze Complexity

Testcase Test Result

Case 1 Case 2 Case 3 +

lists =

[[1,4,5], [1,3,4], [2,6]]

```
29         int index2 = i * 2 + 1;
30         ListNode l1 = lists[index1];
31         ListNode l2 = (index2 < lists.length) ? lists[index2] : null;
32         merged[i] = mergeTwoLists(l1, l2);
33     }
34     lists = merged;
35 }
36
37 return lists[0];
38 }
39 }
```

22:56 06-03-2025