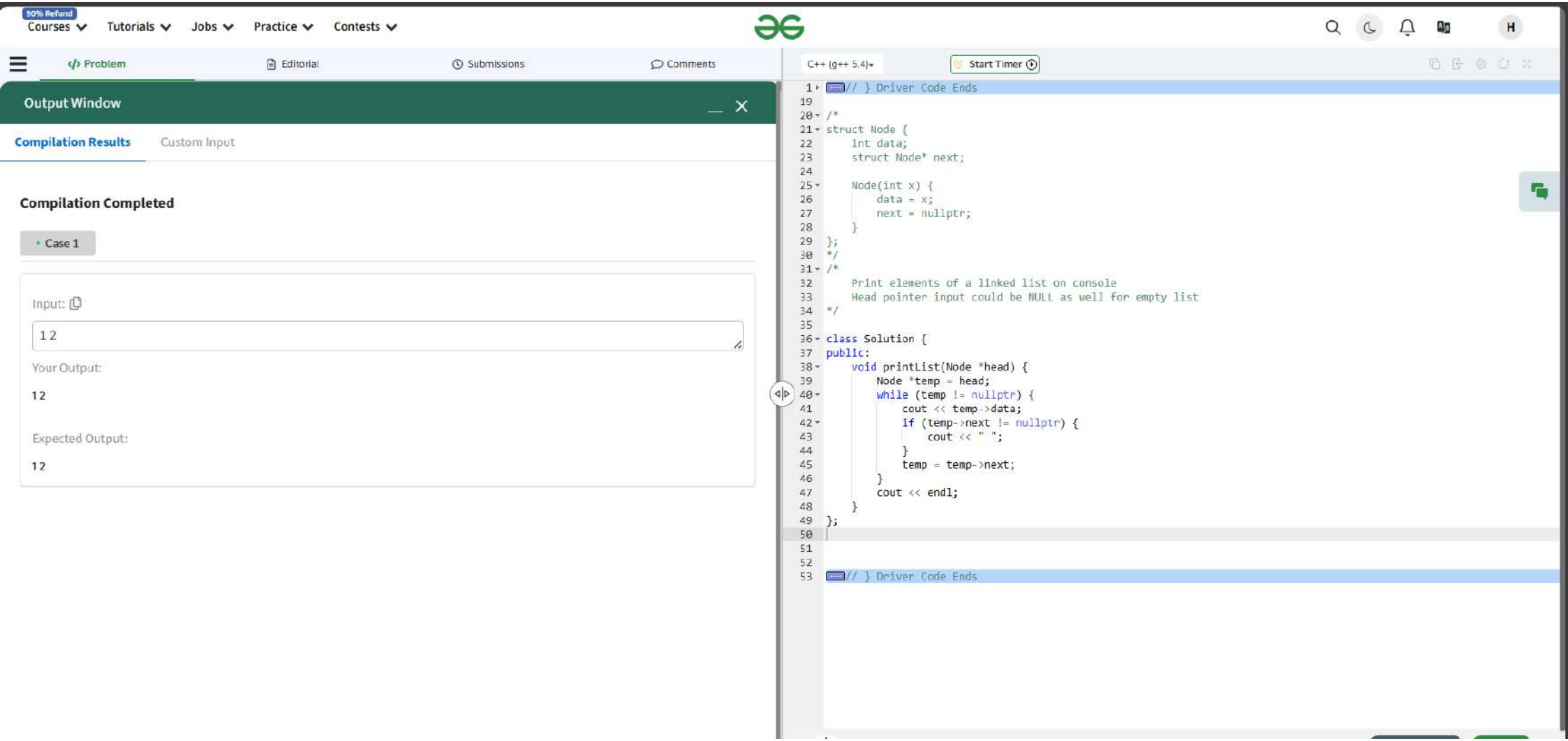

50% Refund
Courses
Tutorials
Jobs
Practice
Contests
Problem
Editorial
Submissions
Comments
Output Window
Compilation Results
Custom Input
Compilation Completed
Case 1
Input:
1 2
Your Output:
1 2
Expected Output:
1 2
C++ (g++ 5.4)
Start Timer
// } Driver Code Ends
/*
struct Node {
int data;
struct Node* next;
Node(int x) {
data = x;
next = nullptr;
}
};
*/
/*
Print elements of a linked list on console
Head pointer input could be NULL as well for empty list
*/
class Solution {
public:
void printList(Node *head) {
Node *temp = head;
while (temp != nullptr) {
cout << temp->data;
if (temp->next != nullptr) {
cout << " ";
}
temp = temp->next;
}
cout << endl;
}
};
// } Driver Code Ends

Problem List

Run | Submit | Premium

Description | Accepted | Editorial | Solutions | Submissions

All Submissions

Accepted 168 / 168 testcases passed

Ashutosh2215 submitted at Mar 06, 2025 20:25

Editorial | Solution

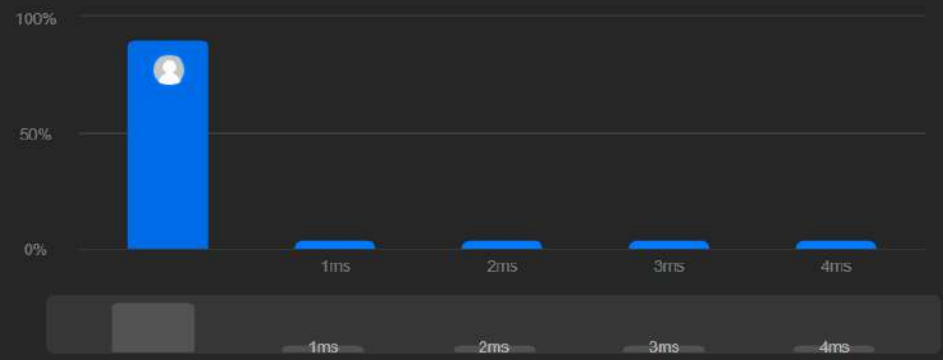
Runtime

0 ms | Beats 100.00%

Analyze Complexity

Memory

16.32 MB | Beats 11.13%



Code | C++

```
/**
 * Definition for singly-linked list.
 * struct ListNode {
 *     int val;
 *     ListNode *next;
 *     ListNode() : val(0), next(nullptr) {}
```

Code

C++ | Auto

```
 */
class Solution {
public:
    ListNode* deleteDuplicates(ListNode* head) {
        ListNode* current = head;

        while (current != nullptr && current->next != nullptr) {
            if (current->val == current->next->val) {

                ListNode* temp = current->next;
                current->next = current->next->next;
                delete temp;
            } else {

                current = current->next;
            }
        }

        return head;
    }
```

Saved | Ln 21, Col 30

Testcase | Test Result

Case 1 | Case 2 | +

head =

[1,1,2]

Source

Problem List | Run | Submit | Premium

Description | Accepted | Editorial | Solutions | Submissions

All Submissions

Accepted 28 / 28 testcases passed

Ashutosh2215 submitted at Mar 06, 2025 20:28

Editorial | Solution

Runtime

0 ms | Beats 100.00%

Analyze Complexity

Memory

13.51 MB | Beats 18.46%

Code | C++

```
/**
 * Definition for singly-linked list.
 * struct ListNode {
 *     int val;
 *     ListNode *next;
 *     ListNode() : val(0), next(nullptr) {}
```

Code

C++ | Auto

```
 * };
 */
class Solution {
public:
    ListNode* reverseList(ListNode* head) {
        ListNode* prev = nullptr;
        ListNode* current = head;
        ListNode* next = nullptr;

        while (current != nullptr) {
            next = current->next;
            current->next = prev;
            prev = current;
            current = next;
        }

        return prev;
    }
};
```

Saved | Ln 25, Col 23

Testcase | Test Result

Case 1 | Case 2 | Case 3

head =

[1,2,3,4,5]

Source

Description | Accepted | Editorial | Solutions | Submissions

All Submissions

Accepted 70 / 70 testcases passed

Ashutosh2215 submitted at Mar 06, 2025 20:30

Editorial | Solution

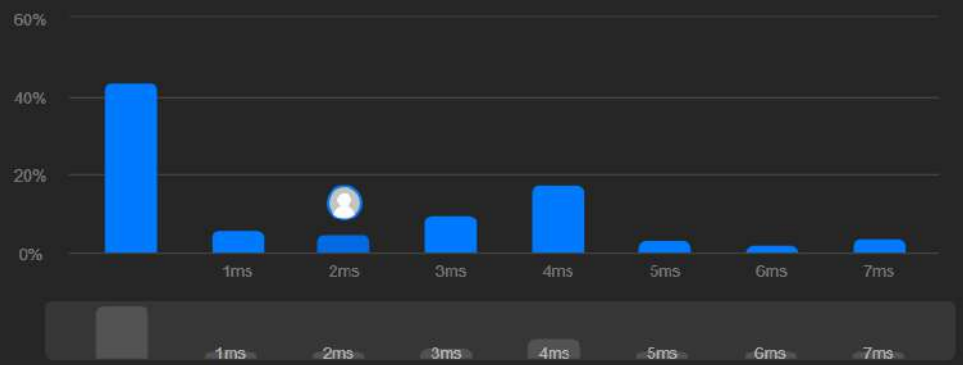
Runtime

2 ms | Beats 50.07%

Analyze Complexity

Memory

312.10 MB | Beats 17.89%



Code | C++

```
/**
 * Definition for singly-linked list.
 * struct ListNode {
 *     int val;
 *     ListNode *next;
 *     ListNode() : val(0), next(nullptr) {}
```

Code

C++ | Auto

```
 */
class Solution {
public:
    ListNode* deleteMiddle(ListNode* head) {
        if (!head || !head->next) return nullptr;

        ListNode* slow = head;
        ListNode* fast = head;
        ListNode* prev = nullptr;

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

        prev->next = slow->next;
        delete slow;

        return head;
    }
```

Saved

Ln 32, Col 1

Testcase | Test Result

Case 1 | Case 2 | Case 3 +

head =

[1,3,4,7,1,2,6]

Source

Problem List | Run | Submit | Premium

Description | Accepted | Editorial | Solutions | Submissions

All Submissions

Accepted 208 / 208 testcases passed

Ashutosh2215 submitted at Mar 06, 2025 20:34

Editorial | Solution

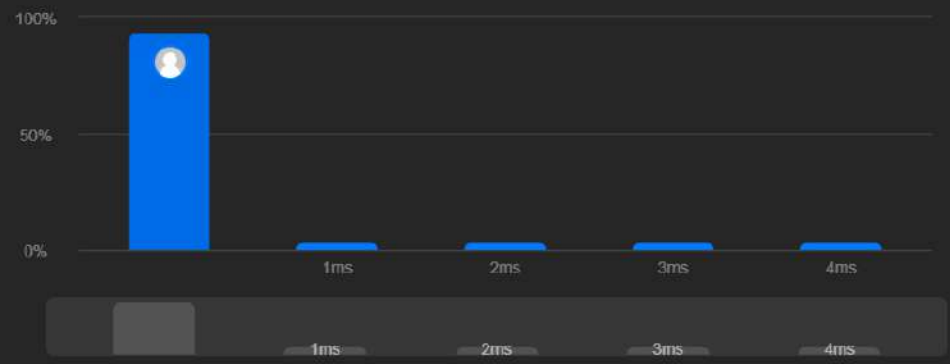
Runtime

0 ms | Beats 100.00%

Analyze Complexity

Memory

19.40 MB | Beats 62.23%



Code | C++

```
/**
 * Definition for singly-linked list.
 * struct ListNode {
 *     int val;
 *     ListNode *next;
 *     ListNode() : val(0), next(nullptr) {}
```

Code

C++ | Auto

```
/**
 * 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) {
        if (!list1) return list2;
        if (!list2) return list1;

        ListNode* dummy = new ListNode(-1);
        ListNode* current = dummy;

        while (list1 && list2) {
```

Saved

Ln 39, Col 1

Testcase | Test Result

Case 1 | Case 2 | Case 3 +

list1 =

[1,2,4]

Source

Proble... Run Submit 1 Premium

Description | Accepted | Editorial | Solutions | Submissions

All Submissions

Accepted 29 / 29 testcases passed

Ashutosh2215 submitted at Mar 06, 2025 20:35

Editorial Solution

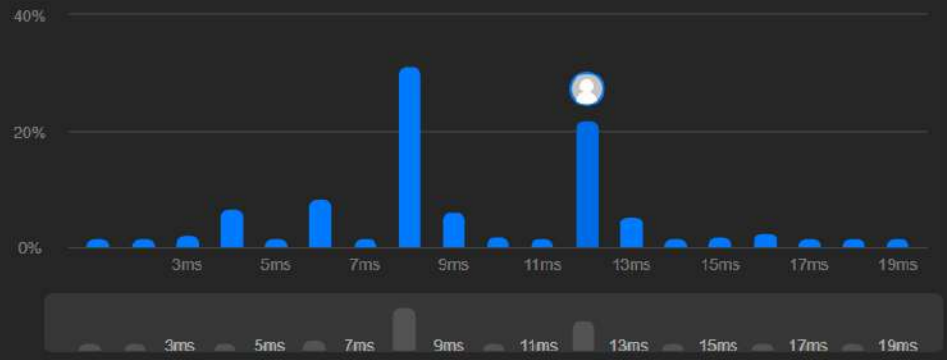
Runtime

12 ms | Beats 40.41%

Analyze Complexity

Memory

11.87 MB | Beats 54.03%



Code | C++

```
/**
 * Definition for singly-linked list.
 * struct ListNode {
 *     int val;
 *     ListNode *next;
 *     ListNode(int x) : val(x), next(NULL) {}
```

Code

C++ Auto

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

        ListNode* slow = head;
        ListNode* fast = head;

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

        return false;
    }
};

```

Saved Ln 26, Col 1

Testcase | Test Result

Case 1 Case 2 Case 3 +

head =

[3,2,0,-4]

Source

Problem List | Run | Submit | Premium

Description | Accepted | Editorial | Solutions | Submissions

All Submissions

Accepted 232 / 232 testcases passed

Ashutosh2215 submitted at Mar 06, 2025 20:36

Editorial | Solution

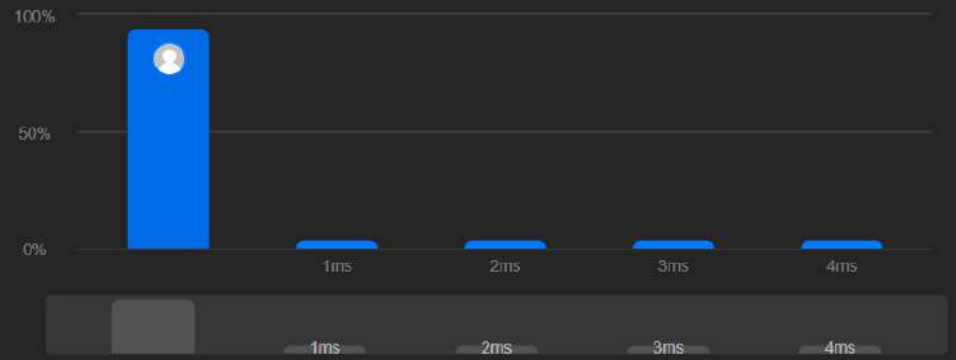
Runtime

0 ms | Beats 100.00%

Analyze Complexity

Memory

16.51 MB | Beats 4.06%



Code | C++

```
/**
 * Definition for singly-linked list.
 * struct ListNode {
 *     int val;
 *     ListNode *next;
 *     ListNode() : val(0), next(nullptr) {}
```

Code

C++ | Auto

```
12  public:
13      ListNode* rotateRight(ListNode* head, int k) {
14          if (!head || !head->next || k == 0) return head;
15
16          ListNode* temp = head;
17          int length = 1;
18
19          while (temp->next) {
20              temp = temp->next;
21              length++;
22          }
23
24          temp->next = head;
25          k = k % length;
26          int stepsToNewHead = length - k;
27
28          while (stepsToNewHead--) {
29              temp = temp->next;
30          }
31
32          head = temp->next;
```

Saved | Ln 38, Col 1

Testcase | Test Result

Case 1 | Case 2 | +

head =

[1,2,3,4,5]

Source

Problem List | Run | Submit | Premium

Description | Accepted | Editorial | Solutions | Submissions

All Submissions

Accepted 30 / 30 testcases passed

Ashutosh2215 submitted at Mar 06, 2025 20:37

Editorial | Solution

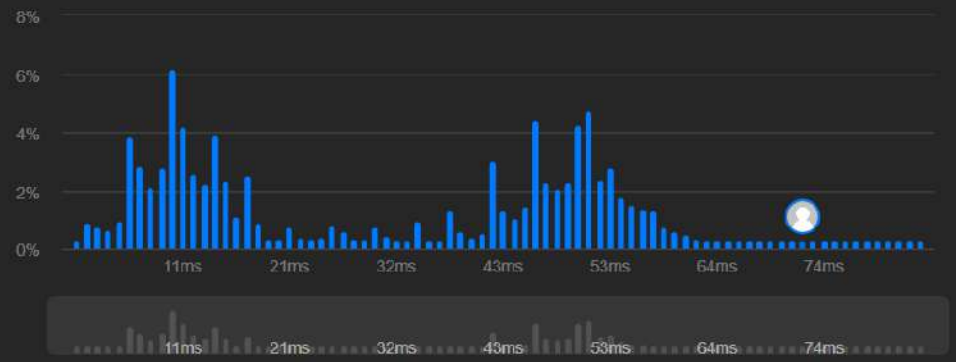
Runtime

72 ms | Beats 6.71%

Analyze Complexity

Memory

86.44 MB | Beats 5.21%



Code | C++

```
/**
 * Definition for singly-linked list.
 * struct ListNode {
 *     int val;
 *     ListNode *next;
 *     ListNode() : val(0), next(nullptr) {}
```

Code

C++ | Auto

```
12  public:
13      ListNode* merge(ListNode* left, ListNode* right) {
14          if (!left) return right;
15          if (!right) return left;
16
17          ListNode* dummy = new ListNode(-1);
18          ListNode* current = dummy;
19
20          while (left && right) {
21              if (left->val <= right->val) {
22                  current->next = left;
23                  left = left->next;
24              } else {
25                  current->next = right;
26                  right = right->next;
27              }
28              current = current->next;
29          }
30
31          if (left) current->next = left;
```

Saved | Ln 64, Col 1

Testcase | Test Result

Case 1 | Case 2 | Case 3 | +

head =

[4,2,1,3]

Source