

DSA Assignment - 24 June

Name - Aditya Tiwari

Roll.No - 23/11/EC/040

LeetCode Id - Aditya1305T

GeekforGeeks - aditya1305t

Github Repo Link -

https://github.com/Aditya1305T/SOE_Training_25

DSA- Section -

Question 1: Design a Singly Linked List and write functions to

- Insert at beginning
- Insert at end
- Insert at any index
- Delete at beginning
- Delete at end
- Delete at index

Submission Screenshot:

```
adityatiwari@Adityas-MacBook-Pro-2 DSA % cd ~/Users/adit
./"q1"
adityatiwari@Adityas-MacBook-Pro-2 output % ./"q1"
6 5
6 5 56 500
100 6 5 56 500
100 6 5 56 200 500 300
5 56 200 500 300
5 56 200
0 1 5 56 200
Please enter valid index.
Index must be from 1 to 4.
0 5 56
Element 65 does not exist in the list.
0 5
adityatiwari@Adityas-MacBook-Pro-2 output %
```

File & Run ⚙️ Compile 🐞 Debug SOE_Training_2025 g++ ▶

Question 2: Delete a Node in Singly Linked List

Platform: GeeksForGeeks

Problem Link:

<https://www.geeksforgeeks.org/problems/delete-a-node-in-single-link>

[ed-list/1](#)

Submission Screenshot:

The screenshot displays a LeetCode submission interface. On the left, the 'Output Window' shows 'Compilation Results' for a submission by 'Y.O.G.I. (AI Bot)'. It indicates 'Problem Solved Successfully' with 2015/2015 test cases passed, 1/1 attempts, 100% accuracy, 2/2 points scored, and a time taken of 0.47 seconds. Below this, there are links to 'Solve Next' problems and a 'Suggested Contest' section for 'Job-A-Thon Hiring Challenge'. On the right, the C++ code for the 'deleteNode' function is shown, which uses a dummy node and a current pointer to traverse the linked list and delete the node at index x-1. The code is as follows:

```
1- /* Link list Node
2- struct Node
3- {
4-     int data;
5-     struct Node* next;
6-
7-     Node(int x){
8-         data = x;
9-         next = NULL;
10-    }
11- };
12- */
13- class Solution {
14- public:
15-     /* Function to delete a node from a linked list */
16-     Node* deleteNode(Node* head, int x) {
17-         Node dummy(-1);
18-         dummy.next = head;
19-         Node* curr = &dummy;
20-         for(int i = 0; i < x-1; i++){
21-             curr = curr->next;
22-         }
23-
24-         curr->next = curr->next->next;
25-         return dummy.next;
26-     }
27- };
```

Question 3: Find Middle

Platform: LeetCode

Problem Link: <https://leetcode.com/problems/middle-of-the-linked-list/>

Submission Screenshot:

Description

Editorial

Solutions

Submissions

Accepted

All Submissions

Accepted

36 / 36 testcases passed

Aditya1305T submitted at Jun 25, 2025 12:38

Editorial

Solution

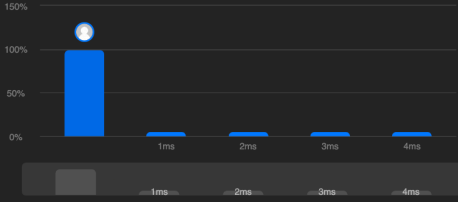
Runtime

0 ms | Beats 100.00%

Analyze Complexity

Memory

10.05 MB | Beats 25.25%



Category	Value
Runtime	0 ms
Memory	10.05 MB

Code | C++

```
/**
 * 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) {}
 * };
 */
```

View more

Write your notes here

C++

Auto

```
1 /**
2  * Definition for singly-linked list.
3  * struct ListNode {
4  *     int val;
5  *     ListNode *next;
6  *     ListNode() : val(0), next(nullptr) {}
7  *     ListNode(int x) : val(x), next(nullptr) {}
8  *     ListNode(int x, ListNode *next) : val(x), next(next) {}
9  * };
10 */
11 class Solution {
12 public:
13     ListNode* middleNode(ListNode* head) {
14         ListNode* slow = head;
15         ListNode* fast = head;
16
17         while(fast && fast->next){
18             slow = slow->next;
19             fast = fast->next->next;
20         }
21
22         return slow;
23     }
24 }
```

Saved

Ln 1, Col 1

Testcase

Test Result

Case 1

Case 2

+

head =

[1,2,3,4,5]

</> Source