H

Practice | 🕐 **Compete** | 💼 **Jobs** | 🎖 **Rank** | 🏆 **Leaderboard**

🔍 [   ]

Points: 25 Rank: 183198

Delete a Node 🔖

H by **harsha_s**

| **Problem** | Submissions | Leaderboard | Discussions | Editorial 🔒 |

This challenge is part of a tutorial track by [MyCodeSchool](#) and is accompanied by a video lesson.

You're given the pointer to the head node of a linked list and the position of a node to delete. Delete the node at the given position and return the head node. A position of 0 indicates head, a position of 1 indicates one node away from the head and so on. The list may become empty after you delete the node.

Input Format

You have to complete the `Node* Delete(Node* head, int position)` method which takes two arguments - the head of the linked list and the position of the node to delete. You should NOT read any input from stdin/console. `position` will always be at least 0 and less than the number of the elements in the list.

Output Format

Delete the node at the given position and `return` the head of the updated linked list. Do NOT print anything to stdout/console.

Sample Input

```
1 --> 2 --> 3 --> NULL, position = 0
1 --> NULL , position = 0
```

Sample Output

```
2 --> 3 --> NULL
NULL
```

Explanation

1. 0th position is removed, 1 is deleted from the list.
2. Again 0th position is deleted and we are left with empty list.

Video lesson

f 𝕏 in

Submissions:[70721](#)

Max Score:5
Difficulty: Easy

Rate This Challenge:

☆ ☆ ☆ ☆ ☆

[More](#)

| Current Buffer (saved locally, editable) ⑂ 🕘 | | C++ ⌄ | ⛶ ⚙ |

```
1  /*
2     Delete Node at a given position in a linked list
3     Node is defined as
```

```
 4   │   struct Node
 5   │   {
 6   │      int data;
 7   │      struct Node *next;
 8   │   }
 9   │*/
10   │Node* Delete(Node *head, int position)
11  ▾│{
12   │  // Complete this method
13   │}
14   │
```

Line: 1 Col: 1

⬆ Upload Code as File   ☐ Test against custom input   Run Code   Submit Code

Join us on IRC at #hackerrank on freenode for hugs or bugs.

Contest Calendar | Blog | Scoring | Environment | FAQ | About Us | Support | Careers | Terms Of Service | Privacy Policy | Request a Feature