





Delete duplicate-value nodes from a sorted linked list

Submissions



Editorial 🖰

This challenge is part of a tutorial track by MyCodeSchool

You are given the pointer to the head node of a sorted linked list, where the data in the nodes is in ascending order. Delete nodes and return a sorted list with each distinct value in the original list. The given head pointer may be null indicating that the list is empty.

Example

Problem

head refers to the first node in the list $1 \rightarrow 2 \rightarrow 2 \rightarrow 3 \rightarrow 3 \rightarrow 3 \rightarrow 3 \rightarrow NULL$.

Remove 1 of the 2 data values and return head pointing to the revised list $1 \rightarrow 2 \rightarrow 3 \rightarrow NULL$.

Leaderboard

Function Description

Complete the removeDuplicates function in the editor below.

removeDuplicates has the following parameter:

• SinglyLinkedListNode pointer head: a reference to the head of the list

Returns

• SinglyLinkedListNode pointer: a reference to the head of the revised list

Input Format

The first line contains an integer t, the number of test cases.

The format for each test case is as follows:

The first line contains an integer $oldsymbol{n}$, the number of elements in the linked list.

Each of the next $m{n}$ lines contains an integer, the $m{data}$ value for each of the elements of the linked list.

Constraints

- $1 \le t \le 10$
- $1 \le n \le 1000$
- $1 \le list[i] \le 1000$

Sample Input

Sample Output

1 2 3 4

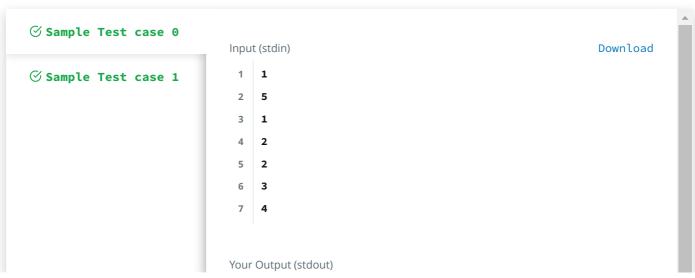
Explanation

```
The initial linked list is: 1	o 2	o 2	o 3	o 4	o NULL. The final linked list is: 1	o 2	o 3	o 4	o NULL.
```

```
Change Theme
                                                       Language Java 8
            JIIIg LYL IIINGUL I JUNOUG 1
68
                int data;
69
                SinglyLinkedListNode next;
70
          * }
71
72
          */
73
74
         public static SinglyLinkedListNode removeDuplicates(SinglyLinkedListNode llist) {
75
         // Write your code here
76
         SinglyLinkedListNode temp = llist;
77
             while(temp.next!=null)
78
79
             {
80
                  if(temp.data == temp.next.data)
81
83
                           temp.next = temp.next.next;
84
                  }
                     else
85
86
87
                        temp = temp.next;
88
89
90
             return llist;
91
92
         }
93
94
                                                                                       Line: 55 Col: 15
                                                                                       Submit Code
                                                                         Run Code
                       Test against custom input
```

Congratulations!

You have passed the sample test cases. Click the submit button to run your code against all the test cases.



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