

Day 24: More Linked Lists



Problem Tutorial Submissions Leaderboard Editorial **Discussions**

Objective

Check out the Tutorial tab for learning materials and an instructional video!

Task

A Node class is provided for you in the editor. A Node object has an integer data field, , and a Node instance pointer, , pointing to another node (i.e.: the next node in a list).

A removeDuplicates function is declared in your editor, which takes a pointer to the node of a

Submitted 32572 times Max Score 30 Need Help?

linked list as a parameter. Complete *removeDuplicates* so that it deletes any duplicate nodes from the list and returns the head of the updated list.

Note: The pointer may be null, indicating that the list is empty. Be sure to reset your pointer when performing deletions to avoid breaking the list.

Input Format

You do not need to read any input from stdin. The following input is handled by the locked stub code and passed to the *removeDuplicates* function:

The first line contains an integer, , the number of nodes to be inserted.

The subsequent lines each contain an integer describing the value of a node being inserted at the list's tail.

Constraints

• The data elements of the linked list argument will always be in non-decreasing order.

Output Format

Your *removeDuplicates* function should return the head of the updated linked list. The locked stub code in your editor will print the returned list to stdout.

Sample Input

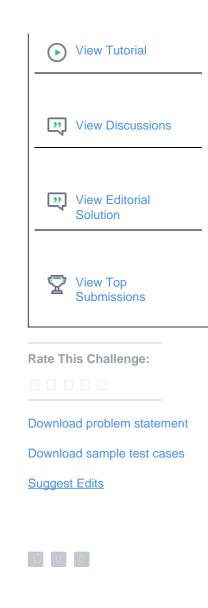


Sample Output

```
1 2 3 4
```

Explanation

, and our non-decreasing list is . The values and both occur twice in the list, so we remove the two duplicate nodes. We then return our updated (ascending) list, which is .





```
11
                    head.next=p
12
               else:
                   start=head
13
14
                    while(start.next!=None):
15
                        start=start.next
16
                    start.next=p
17
                return head
18
       def display(self,head):
19
           current = head
20
            while current:
21
                print(current.data,end=' ')
22
                current = current.next
23
        def removeDuplicates(self,head):
24
            #Write your code here
25
           if head == None:
26
                return None
27
            cur = head
28 🗌
            while cur.next:
29
                if cur.data == cur.next.data:
30
                    cur.next = cur.next.next
31
                else:
32
                    cur = cur.next
33
           return head
34
35
36
37
38
39
40
41
42
43
44
45 mylist= Solution()
46 T=int(input())
47 head=None
48 \square for i in range(T):
49
       data=int(input())
       head=mylist.insert(head,data)
50
51 head=mylist.removeDuplicates(head)
52 mylist.display(head);
```

Line: 25 Col: 25

Upload Code as File Test against custom input Run Code

Congrats, you solved this challenge! Challenge your friends:		
☐ Test Case #0☐ Test Case #3	☐ Test Case #1 ☐ Test Case #4	☐ Test Case #2
Next Challenge You've earned 30.00 points. You are now 5 challenges away from the gold level for your 30 days of code badge.		

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