Your Day 24: More Linked Lists submission got 30.00 points. Tweet

You are now 5 challenges away from the gold level for your 30 days of code badge.

Try the Next Challenge | Try a Random Challenge

Day 24: More Linked Lists

by harsha_s

Problem Submissions Leaderboard Discussions Editorial Tutorial

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

```
6
1
2
2
3
3
4
```

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 .

View Tutorial

View Discussions

View Editorial Solution

View Top Submissions

Rate This Challenge:

Download problem statement

Download sample test cases

Suggest Edits

Current Buffer (saved locally, editable) Python 3

```
class Node:↔
class Solution:
    def insert(self,head,data):
            p = Node(data)
            if head==None:
                head=p
            elif head.next==None:
```

```
                head.next=p
            else:
                start=head
                while(start.next!=None):
                    start=start.next
                start.next=p
            return head
    def display(self,head):
        current = head
        while current:
            print(current.data,end=' ')
            current = current.next
```

```
    def removeDuplicates(self,head):
        #Write your code here
        if head == None:
            return None
        cur = head
        while cur.next:
            if cur.data == cur.next.data:
                cur.next = cur.next.next
            else:
                cur = cur.next
        return head
```

```
mylist= Solution()
T=int(input())
head=None
for i in range(T):
    data=int(input())
    head=mylist.insert(head,data)
head=mylist.removeDuplicates(head)
mylist.display(head);
```

Line: 25 Col: 25

Test against custom input

Run Code Submit Code

Congrats, you solved this challenge!

Challenge your friends:

- Test Case #0
- Test Case #1
- Test Case #2
- Test Case #3
- Test Case #4

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.