



Practice ▾ Tutorials ▾ 30 Days of Code ▾ Day 15: Linked List

COMPETE

JOBS

LEADERBOARD

6 more challenges to get your next star!

Days of  
Code

0%

1

joel\_h\_healy ▾

Your Day 15: Linked List submission got 30.00 points. Share

You are now 6 challenges away from the 4th star for your 30 days of code badge.

[Try the Next Challenge](#) | [Try a Random Challenge](#)

# Day 15: Linked List ▾

by [harsha\\_s](#)

Problem

Submissions

Leaderboard

Discussions

Editorial ▾

Tutorial

## Objective

Today we're working with *Linked Lists*. Check out the [Tutorial](#) tab for learning materials and an instructional video!

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

Submitted 56908 times  
Max Score 30

Need Help?

A *Node insert* function is also declared in your editor. It has two parameters: a pointer, , pointing to the first node of a linked list, and an integer value that must be added to the end of the list as a new *Node* object.

Task

Complete the *insert* function in your editor so that it creates a new *Node* (pass as the *Node* constructor argument) and inserts it at the tail of the linked list referenced by the parameter. Once the new node is added, return the reference to the node.

**Note:** If the argument passed to the *insert* function is *null*, then the initial list is empty.

Input Format

The *insert* function has parameters: a pointer to a *Node* named , and an integer value, . The constructor for *Node* has parameter: an integer value for the field.

You *do not* need to read anything from stdin.

Output Format

Your *insert* function should return a reference to the node of the linked list.

Sample Input

The following input is handled for you by the locked code in the editor:  
The first line contains *T*, the number of test cases.  
The subsequent lines of test cases each contain an integer to be inserted at the list's tail.

```
4
2
3
4
1
```

Sample Output

The locked code in your editor prints the ordered data values for each element in your list as a single line of space-separated integers:

```
2 3 4 1
```

Explanation

, so the locked code in the editor will be inserting nodes.  
The list is initially empty, so is null; accounting for this, our code returns a new node containing the data value as the of our list. We then create and insert nodes , , and at the tail of our list. The resulting list returned by the last call to is , so the printed output is 2 3 4 1 .

 [View Tutorial](#)

 [View Discussions](#)

 [View Editorial Solution](#)

 [View Top Submissions](#)

Rate This Challenge:

☐☐☐☐☐

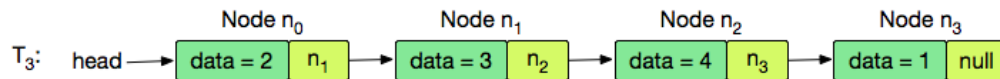
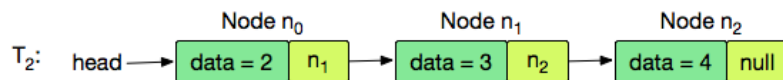
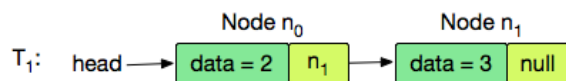
[Download problem statement](#)

[Download sample test cases](#)

[Suggest Edits](#)



Initial: head → null



Current Buffer (saved locally, editable) ☐ ☐

Python 3 ☐ ☐ ☐

```
1 class Node:
2     def __init__(self, data):
3         self.data = data
4         self.next = None
5 class Solution:
6     def display(self, head):
7         current = head
8         while current:
9             print(current.data, end=' ')
10            current = current.next
```


```
11 def insert(self, head, data):
12     #Complete this method
13     new_node = Node(data)
14     if not head:
15         return new_node
16     else:
17         current = head
18         while current.next:
19             current = current.next
20         current.next = new_node
21     return head
22
23
```

```
24 mylist= Solution()
25 T=int(input())
26 head=None
27 for i in range(T):
28     data=int(input())
29     head=mylist.insert(head,data)
30 mylist.display(head);
```

☐ [Upload Code as File](#)

[Run](#) [Code](#)

**Congrats, you solved this challenge!**

Challenge your friends:   

☒ Test Case #0

☐ Test Case #1

☐ Test Case #2

**You've earned 30.00 points. You are now 6 challenges away from the 4th star for your 30 days of code badge.**

[Next Challenge](#)