



Find Merge Point of Two Lists ★

234 more points to get your next star!

Rank: 904751 | Points: 241/475



Your Find Merge Point of Two Lists submission got 5.00 points.

Share

Post



You are now 234 points away from the 4th star for your problem solving badge.

[Try the next challenge](#)

Problem

Submissions

Leaderboard

Editorial

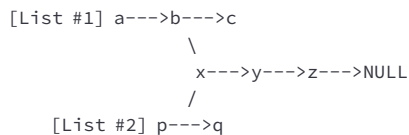
This challenge is part of a tutorial track by [MyCodeSchool](#)

Given pointers to the head nodes of **2** linked lists that merge together at some point, find the node where the two lists merge. The merge point is where both lists point to the same node, i.e. they reference the same memory location. It is guaranteed that the two head nodes will be different, and neither will be NULL. If the lists share a common node, return that node's **data** value.

Note: After the merge point, both lists will share the same node pointers.

Example

In the diagram below, the two lists converge at Node x:



Function Description

Complete the findMergeNode function in the editor below.

findMergeNode has the following parameters:

- SinglyLinkedListNode pointer head1: a reference to the head of the first list
- SinglyLinkedListNode pointer head2: a reference to the head of the second list

Returns

- int: the **data** value of the node where the lists merge

Input Format

Do not read any input from stdin/console.

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

Each of the test cases is in the following format:

The first line contains an integer, **index**, the node number where the merge will occur.

The next line contains an integer, **list1_count** that is the number of nodes in the first list.

Each of the following **list1_count** lines contains a **data** value for a node. The next line contains an integer, **list2_count** that is the number of nodes in the second list.

Each of the following **list2_count** lines contains a **data** value for a node.

Constraints

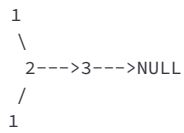
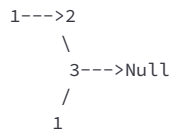
The lists will merge.

head1, head2 \neq null.

head1 \neq **head2**.

Sample Input

The diagrams below are graphical representations of the lists that input nodes **head1** and **head2** are connected to.

Test Case 0**Test Case 1****Sample Output**

```

2
3

```

Explanation

Test Case 0: As demonstrated in the diagram above, the merge node's data field contains the integer **2**.

Test Case 1: As demonstrated in the diagram above, the merge node's data field contains the integer **3**.

[Change Theme](#)

Language

Python 3



```

40
41 #
42 # For your reference:
43 #
44 # SinglyLinkedListNode:
45 #     int data
46 #     SinglyLinkedListNode next
47 #
48 #
49 def findMergeNode(head1, head2):
50     length1, length2 = 0, 0
51     cur1, cur2 = head1, head2
52     # Find length
53     while cur1:
54         length1 += 1
55         cur1 = cur1.next
56     while cur2:
57         length2 += 1
58         cur2 = cur2.next
59
60     cur1, cur2 = head1, head2
61     if length1 < length2:
62         for _ in range(length2 - length1):
63             cur2 = cur2.next
64     else:
65         for _ in range(length1 - length2):
66             cur1 = cur1.next

```

```
66         cur1 = cur1.next
67     while cur1:
68         if cur1 == cur2:
69             return cur1.data
70         cur1 = cur1.next
71         cur2 = cur2.next
72
73 if name == 'main': ...
```

EMACS

Line: 41 Col: 1

⬆️ Upload Code as File

☐ Test against custom input

Run Code


Submit Code

You have earned 5.00 points!

You are now 234 points away from the 4th star for your problem solving badge.

15%

241/475



Congratulations

You solved this challenge. Would you like to challenge your friends?


Next Challenge

✔️ Test case 0

Compiler Message

Success

✔️ Test case 1

✔️ Test case 2 

Input (stdin)

11

21

33


41


52


63


71

81

✔️ Test case 3 

✔️ Test case 4 

✔️ Test case 5 

✔️ Test case 6 

Download