Find Merge Point of Two Lists ☆

Problem   Submissions   Leaderboard   Editorial

RATE THIS CHALLENGE

☆ ☆ ☆ ☆ ☆

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:

```
    [List #1] a--->b--->c
                         \
                          x--->y--->z--->NULL
                         /
        [List #2] p--->q
```
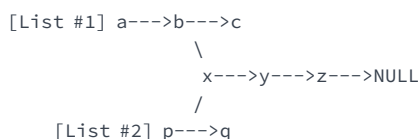
Function Description

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

```
  1
   \
    2--->3--->NULL
   /
  1
```

Test Case 1

```
  1--->2
      \
       3--->Null
      /
     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   C++

```cpp
66    *       int data;
67    *       SinglyLinkedListNode* next;
68    * };
69    *
70    */
71    int findMergeNode(SinglyLinkedListNode* head1, SinglyLinkedListNode* head2) {
72        SinglyLinkedListNode* temp1 = head1;
73
74        while(temp1!= NULL){
75            SinglyLinkedListNode* temp2 = head2;
76            while(temp2 !=  NULL){       //looping through both lists
77                if(temp1 == temp2){   // matching common memory location
78                    return temp2->data;
79                }
80                temp2= temp2->next;
81            }
82            temp1 = temp1->next;
83        }
84        return NULL;
85    }
86
87    int main() …
```

Line: 71 Col: 1

⬆ Upload Code as File   ☐ Test against custom input

Run Code   Submit Code

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

https://www.hackerrank.com/challenges/find-the-merge-point-of-two-joined-linked-lists/problem?h_l=interview&playlist_slugs%5B%5D=interview-prepa…   2/2