



Description

Solution

Discuss (999+)

Submissions

2. Add Two Numbers

Medium

15707

3389

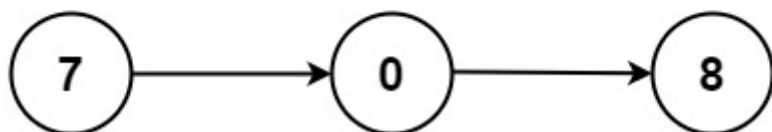
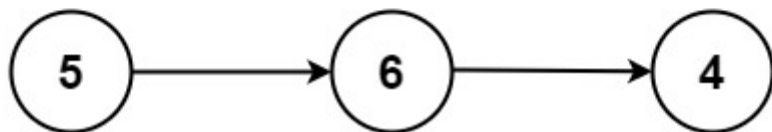
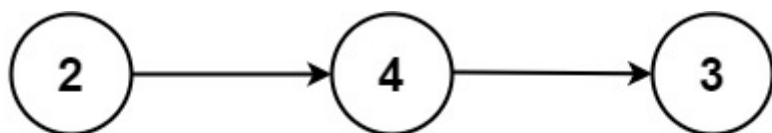
Add to List

Share

You are given two **non-empty** linked lists representing two non-negative integers. The digits are stored in **reverse order**, and each of their nodes contains a single digit. Add the two numbers and return the sum as a linked list.

You may assume the two numbers do not contain any leading zero, except the number 0 itself.

Example 1:



Input: l1 = [2,4,3], l2 = [5,6,4]

Output: [7,0,8]

Explanation: 342 + 465 = 807.

Example 2:

Input: l1 = [0], l2 = [0]

Output: [0]

Example 3:

C#

i

{}

↶

↷

```
1  /**
2   * Definition for singly
3   * linked list.
4   * public class ListNode
5   * {
6   *     public int val;
7   *     public ListNode
8   *     next;
9   *     public ListNode(
10    *     int val, ListNode
11    *     next) {
12    *         this.val = val;
13    *         this.next =
14    *         next;
15    *     }
16    * }
17    */
18    public class Solution {
19    *     public ListNode
20    *     AddTwoNumbers(ListNode
21    *     l1, ListNode l2) {
22    *
23    *         var head = new
24    *         ListNode(0);
25    *         var current = head;
26    *         var carry = 0;
27    *         while(l1 != null || l2 != null || carry != 0)
28    *         {
29    *             int x = l1 != null ? l1.val : 0;
30    *             int y = l2 != null ? l2.val : 0;
31    *             int sum = x + y + carry;
32    *             carry = sum / 10;
33    *             current.next = new ListNode(sum % 10);
34    *             current = current.next;
35    *             if (l1 != null) l1 = l1.next;
36    *             if (l2 != null) l2 = l2.next;
37    *         }
38    *         return head.next;
39    *     }
40    * }
```

Your previous code was restored from y

Testcase

Run Code Result

[2,4,3]

[5,6,4]

Console

Use Example Testcase

Problems

Pick One

< Prev

2/2136

Next >

Run Code ^

Submit