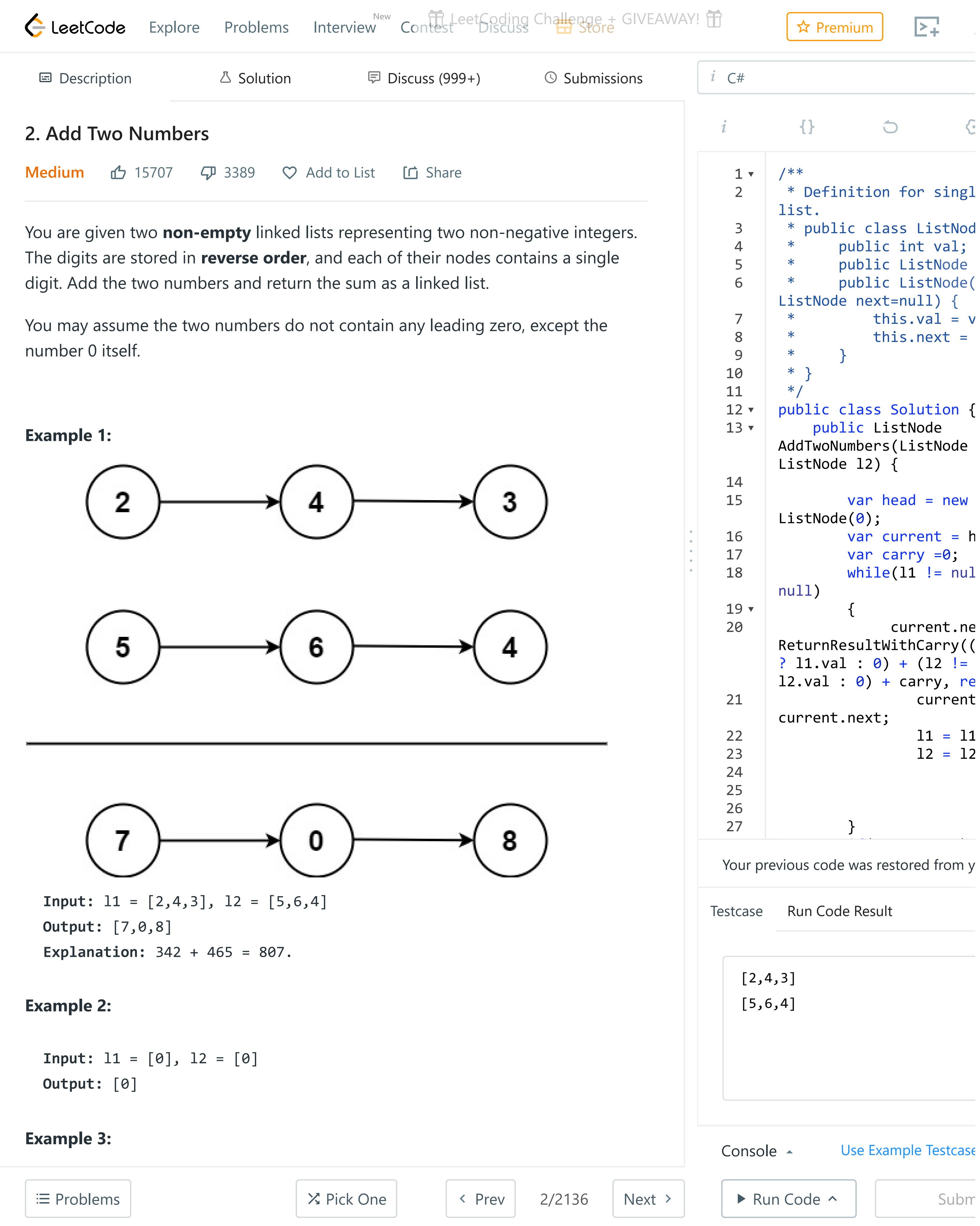

2. Add Two Numbers

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:

```
/**
 * Definition for singl
list.
 * public class ListNod
 *     public int val;
 *     public ListNode
 *     public ListNode(
ListNode next=null) {
 *         this.val = v
 *         this.next =
 *     }
 * }
 */
public class Solution {
    public ListNode
AddTwoNumbers(ListNode
ListNode l2) {

        var head = new
ListNode(0);
        var current = h
        var carry =0;
        while(l1 != nul
null)
        {
             current.ne
ReturnResultWithCarry((
? l1.val : 0) + (l2 !=
l2.val : 0) + carry, re
                current
current.next;
                l1 = l1
                l2 = l2



        }
```

Your previous code was restored from y

Testcase Run Code Result

```
[2,4,3]
[5,6,4]
```