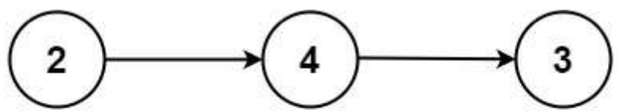
2. Add Two Numbers

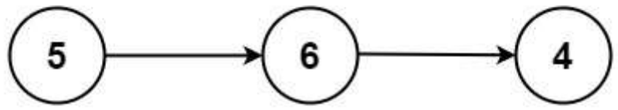
Medium 17487 3630 Add to List Share

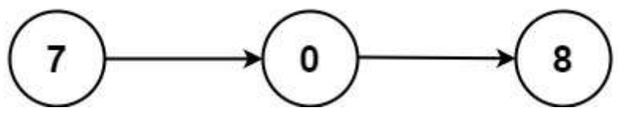
You are given two **non-empty** linked lists representing two non-negative integers. The digits are stored in **reverse order**, and each of their nodes contains a single digit. Add the two numbers and return the sum as a linked list.

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:

Java Autocomplete

```java
/**
 * Definition for singly-linked list.
 * public class ListNode {
 *     int val;
 *     ListNode next;
 *     ListNode() {}
 *     ListNode(int val) { this.val = val; }
 *     ListNode(int val, ListNode next) { this.val = val; this.next = next; }
 * }
 */
class Solution {

    public ListNode addTwoNumbers(ListNode l1, ListNode l2) {

        int carry = 0;
        ListNode listNode = new ListNode();
        ListNode currentNode = listNode;

        while(l1 != null || l2 != null){

            int sum = 0 + carry;

            if(l1 != null){
                sum += l1.val;
                l1 = l1.next;
            }

            if(l2 != null){
                sum += l2.val;
                l2 = l2.next;
            }

            currentNode.next = new ListNode(sum % 10);
            carry = sum / 10;
            currentNode = currentNode.next;
        }

        if(carry != 0)
            currentNode.next = new ListNode(carry);


        return listNode.next;
    }

}
```