2. Add Two Numbers

Easy,
Linked List, Math.

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 contain a single digit. Add the two numbers and return it as a linked list.

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

Example:

```
Input: (2 -> 4 -> 3) + (5 -> 6 -> 4)
Output: 7 -> 0 -> 8
Explanation: 342 + 465 = 807.
```

解法

java

```java
/**
 * Definition for singly-linked list.
 * public class ListNode {
 *     int val;
 *     ListNode next;
 *     ListNode(int x) { val = x; }
 * }
 */
class Solution {
    public ListNode addTwoNumbers(ListNode l1, ListNode l2) {
        ListNode tmp = new ListNode(-1);
        ListNode res = tmp;
        int forAdd=0;
        while(l1!=null || l2!=null){
```

```
            if(l1!=null && l2 !=null){
                int sum = l1.val + l2.val;
                tmp.next = new ListNode((sum+forAdd)%10);
                tmp=tmp.next;
                forAdd=(sum+forAdd)/10;
                l1 = l1.next;
                l2 = l2.next;
            }
            if(l1!=null && l2==null){
                tmp.next = new ListNode((l1.val+forAdd)%10);
                tmp=tmp.next;
                forAdd=(l1.val+forAdd)/10;
                l1 = l1.next;
            }
            if(l1==null && l2!=null){
                tmp.next = new ListNode((l2.val+forAdd)%10);
                tmp = tmp.next;
                forAdd=(l2.val+forAdd)/10;
                l2 = l2.next;
            }
        }
        if(forAdd>0) tmp.next = new ListNode(forAdd);
        return res.next;
    }
}
```