2. Add Two Numbers

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.

Input: (2 -> 4 -> 3) + (5 -> 6 -> 4)
Output: 7 -> 0 -> 8

两个链表相加的问题，链表是 reverse order， 所以应该是 342+465=807.

注意跟踪进位 keep track of carry:

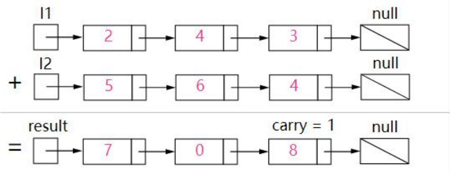

Figure 1. Visualization of the addition of two numbers: $342 + 465 = 807$. *Each node contains a single digit and the digits are stored in reverse order.*

```
class Solution(object):
    def addTwoNumbers(self, l1, l2):
        head=temp=ListNode(0)      #将当前节点初始化为返回列表的虚拟头。
        carry=0    #初始化进位为 0
        while(l1 or l2):
            temp1=l1.val if l1 else 0    #如果已经到 l1 的末端，设置为 0
            temp2=l2.val if l2 else 0
            val=temp1+temp2+carry
            carry= val/10          #更新进位的值
            val=val%10    #创建一个数值为新节点（val mod 10）并将其设置为当前节点
                             #下一个节点，然后将当前节点推进到下一个节点
            temp.next= ListNode(val)    #检测表头的下一个节点
            temp= temp.next
            if l1:
                l1=l1.next     #移动表 1
            if l2:
                l2=l2.next    #移动表 2
        if carry:
            temp.next= ListNode(carry)    #检测表头的下一个节点
            temp=temp.next
        return head.next
```