**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.

**Example:**

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

**分析：**

根据样例可以得知，题中的链表不带头结点。对于无头结点的链表，将第一个节点特殊处理

class Solution {  
public:  
 ListNode\* addTwoNumbers(ListNode\* l1, ListNode\* l2) {  
 ListNode \*res = new ListNode((l1->val + l2->val) % 10);  
 int carry = (l1->val + l2->val) / 10;  
 l1 = l1->next, l2 = l2->next;  
 ListNode \*cur = res;  
 while(l1 != NULL || l2 != NULL){  
 int temp = carry;  
 if(l1){  
 temp += l1->val;  
 l1 = l1->next;  
 }  
 if(l2){  
 temp += l2->val;  
 l2 = l2->next;  
 }  
 carry = temp / 10;  
 temp %= 10;  
 cur->next = new ListNode(temp);  
 cur = cur->next;  
 }  
 if(carry != 0){  
 cur->next = new ListNode(carry);  
 }  
 return res;  
 }  
};

1. new的用法
2. 注意循环中的条件判断，应该是当前节点不空时继续，而不是后继不为空