题目2：

给定两个非空链表来表示两个非负整数。位数按照逆序方式存储，它们的每个节点只存储单个数字。将两数相加返回一个新的链表。

你可以假设除了数字 0 之外，这两个数字都不会以零开头。

思路：

错误思路：把两个链表各转换为数字，并相加写进第三个链表。

错因：题目没有说两个数字的范围，因此不能用这种方法，数字够大时会溢出。

思路：两个链表各取最后的数字，相加结果对10求余写进新链表，并保存下结果除10，然后两个链表分别右移，对计算结果加上上一轮保存的结果。直到两个链表都结束。然后如果上一个结果除0不为0，就再把这个结果加入链表。

源码：

思路1：

class Solution {

public:

ListNode \* addTwoNumbers(ListNode\* l1, ListNode\* l2) {

ListNode\* res;

ListNode ret(-1);

res = &ret;

//cout << ret->val << endl;

int n1 = 0;

int n2 = 0;

int num = 0;

int tmp = 0;

while (l1||l2) {

if (l1){

n1 = l1->val;

l1 = l1->next;

}

else n1 = 0;

if (l2) {

n2 = l2->val;

l2 = l2->next;

}

else n2 = 0;

num = n1 + n2 + tmp;

tmp = num / 10;

ListNode\* t = new ListNode(num % 10);

res->next = t;

//cout << ret.val << endl;

//if (ret.val == 0) {

// ret = res;

//}

//cout << res.val << endl;

res = res->next;

}

if (tmp != 0) {

ListNode\* t = new ListNode(tmp);

res->next = t;

//cout << res->val << endl;

//res = \*res.next;

}

//while (ret.next) {

// cout << ret.val << endl;

// ret = \*ret.next;

//}

return ret.next;

}

};

错误思路：

class Solution {

public:

ListNode \* addTwoNumbers(ListNode\* l1, ListNode\* l2) {

ListNode\* res;

ListNode\* ret;

//cout << ret->val << endl;

long long int n1 = 0;

long long int n2 = 0;

long long int tmp = 1;

while (l1) {

n1 += tmp \* l1->val;

l1 = l1->next;

tmp \*= 10;

}

tmp = 1;

while (l2) {

n2 += tmp \* l2->val;

l2 = l2->next;

tmp \*= 10;

}

long long int num = n1 + n2;

cout << n1 <<" "<< n2 << " " << num << endl;

ListNode \*t0 = new ListNode(num % 10);

res = t0;

ret = res;

num /= 10;

while (num >= 1) {

ListNode \*t = new ListNode(num % 10);

res->next = t;

res = res->next;

num /= 10;

}

return ret;

}

};