2. Add Two Numbers

Medium

44791103FavoriteShare

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.

CPP

/\*\*

\* Definition for singly-linked list.

\* struct ListNode {

\* int val;

\* ListNode \*next;

\* ListNode(int x) : val(x), next(NULL) {}

\* };

\*/

class Solution {

public:

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

ListNode anchor(0);

ListNode\* curr = &anchor;

int carry=0;

while(l1!=NULL||l2!=NULL||carry==1){

int v1=0,v2=0;

if(l1!=NULL){

v1=l1->val;

l1=l1->next;

}

if(l2!=NULL){

v2=l2->val;

l2=l2->next;

}

int val=v1+v2+carry;

ListNode\* n=new ListNode(val);

curr->next=n;

curr=n;

//curr->next=NULL;

curr->val=val%10;

carry=val>9;

//curr->next=NULL;

}

curr->next=NULL;

return anchor.next;

}

};

Success

[Details](https://leetcode.com/submissions/detail/207465460/)

Runtime: 48 ms, faster than 77.95% of C++ online submissions for Add Two Numbers.

Memory Usage: 19.2 MB, less than 0.86% of C++ online submissions forAdd Two Numbers.