445. Add Two Numbers II

Add to List | Question | Editorial Solution | My Submissions (/problems/add-two-numbers-ii/submissions/)

Total Accepted: **9140** Total Submissions: **19782** Difficulty: **Medium** Contributors: **Admin**

You are given two **non-empty** linked lists representing two non-negative integers. The most significant digit comes first 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.

Notes

Follow up:

What if you cannot modify the input lists? In other words, reversing the lists is not allowed.

Example:

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

Subscribe (/subscribe/) to see which companies asked this question

Hide Tags | Linked List (/tag/linked-list/)

Hide Similar Problems | (M) Add Two Numbers (/problems/add-two-numbers/)

Have you met this question in a real interview? Yes No

Discuss (https://discuss.leetcode.com/category/571) | Top Solutions | Pick One (/problems/random-one-question/)

C++

```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) {
        if(l1==NULL) return l2;
        if(l2==NULL) return l1;
        return reverse(add(reverse(l1),reverse(l2)));
    }

    ListNode* reverse(ListNode* l){
        ListNode* cur=l;
        ListNode* head=NULL;
        ListNode* next;
        while(cur != NULL){
            next = cur->next;
            cur->next = head;
            head = cur;
            cur = next;
        }
        return head;
    }

    ListNode* add(ListNode* l1,ListNode* l2){
        ListNode* dummy = new ListNode(0);
        dummy->next = NULL;
        ListNode* cur = dummy;
        int carrier = 0;
        int val;
        while(l1!=NULL&&l2!=NULL){
            val = carrier + l1->val + l2->val;
            l1->val = val%10;
            carrier = val/10;
            cur->next = l1;
            l1 = l1->next;
            l2 = l2->next;
```

```
            cur = cur->next;
        }
        if(l1!=NULL){
            while(l1!=NULL){
                val = carrier + l1->val;
                l1->val = val%10;
                carrier = val/10;
                cur->next = l1;
                l1 = l1->next;
                cur = cur->next;
                if(carrier==0) break;
            }
        }
        else if(l2!=NULL){
            while(l2!=NULL){
                val = carrier + l2->val;
                l2->val = val%10;
                carrier = val/10;
                cur->next = l2;
                l2 = l2->next;
                cur = cur->next;
                if(carrier==0) break;
            }
        }

        if(carrier!=0){
            cur->next = new ListNode(carrier);
        }
        return dummy->next;
    }
};
```

Notes

Custom Testcase

Contribute Testcase

Run Code | Submit Solution