

# 002. Add Two Numbers

## 002 Add Two Numbers

- Linked List+Math

### Description

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.

### 1. Thought Line

#### (1) Discuss how to reverse a linked list

```
1 ListNode* reverseListOrder(ListNode* lst){
2     ListNode* dummyHead = new ListNode(0);
3     ListNode* dummyTail = nullptr;
4     ListNode* insertTo = dummyTail;
5     while(lst!=nullptr){
6         ListNode* insertNode = new ListNode(lst->val);
7         insertNode->next = insertTo;
8         dummyHead->next = insertNode;
9         insertTo = insertNode;
10        lst = lst->next;
11    }
12    return dummyHead->next;
13 }
```

#### (2) Math logic

- Need to analyze the carry direction.
- in "2 -> 4 -> 3"+"5 -> 6 -> 4", it is from left to right
- in "342"+"465", it is from right to left

### 2. Linked List + Math

```
1 /**
2  * Definition for singly-linked list.
3  * struct ListNode {
4  *     int val;
5  *     ListNode *next;
6  *     ListNode(int x) : val(x), next(NULL) {}
7  * };
8  */
9 class Solution {
10
11 private:
12     ListNode* addList(ListNode* l1, ListNode* l2){
13         ListNode* dummyHead = new ListNode(0);
14         ListNode* head = dummyHead;
15         signed int carry = 0;
```

```

16 while(l1!=nullptr || l2!=nullptr || carry !=0){
17     int l1Val = (l1!=nullptr) ? l1->val : 0;
18     int l2Val = (l2!=nullptr) ? l2->val : 0;
19     int sumOfl1l2carry = l1Val + l2Val + carry;
20
21     head->next = new ListNode (sumOfl1l2carry%10);
22     carry = sumOfl1l2carry/10;
23
24     l1 = (l1!=nullptr)? l1->next : nullptr;
25     l2 = (l2!=nullptr)? l2->next : nullptr;
26     head = head->next;
27 }
28 return dummyHead->next;
29 }
30
31 public:
32 ListNode* addTwoNumbers(ListNode* l1, ListNode* l2) {
33     if (l1==nullptr) return l2;
34     if (l2==nullptr) return l1;
35     ListNode* l1Addl2 = addList(l1,l2);
36     return l1Addl2;
37 }
38 };

```