Problem 2 “Add Two Numbers”

Objective: 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.

# Definition for singly-linked list.

# class ListNode:

# def \_\_init\_\_(self, x):

# self.val = x

# self.next = None

class Solution:

def addTwoNumbers(self, l1, l2):

"""

:type l1: ListNode

:type l2: ListNode

:rtype: ListNode

"""

ant = []

while(l1 is not None and l2 is not None):

ans = l1.val + l2.val

if(ans >= 10 or ans <= -10):

ant.append(ans%10)

if(l2.next is not None):

l2.next.val += 1

elif(l1.next is not None):

l1.next.val +=1

else:

ant.append(1)

else:

ant.append(ans)

l1 = l1.next

l2 = l2.next

while(l1 is not None):

if(l1.val >= 10):

ant.append(l1.val%10)

if(l1.next is None):

ant.append(1)

else:

l1.next.val +=1

else:

ant.append(l1.val)

#ant[-1] += l1.val

l1 = l1.next

while(l2 is not None):

if(l2.val >= 10):

ant.append(l2.val%10)

if(l2.next is None):

ant.append(1)

else:

l2.next.val +=1

else:

ant.append(l2.val)

l2 = l2.next

return ant

