**SUM LIST**

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.

class Solution **{**

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

int carry **=** 0**;**

ListNode head **=** **new** ListNode**(-**1**);**

ListNode runner **=** head**;**

**while(**l1 **!=** null **||** l2 **!=** null **||** carry **!=** 0**){** //Check if there is something to sum

int a **=**0**;**

int b **=**0**;**

**if(**l1 **!=** null**)**

a **=** l1**.**val**;**

**if(**l2 **!=** null**)**

b **=** l2**.**val**;**

int num **=** a **+** b **+** carry**;**

carry **=** num**/**10**;** //Get the new carry

num **%=**10**;** //get the next number

runner**.**next **=** **new** ListNode**(**num**);**

runner **=** runner**.**next**;**

l1 **=** l1 **==** null **?** null**:**l1**.**next**;**

l2 **=** l2 **==** null **?** null**:**l2**.**next**;**

**}**

**return** head**.**next**;** //Remove dummyhead

**}**

**}**