#create node with value and next -> none  
class ListNode:  
 def \_\_init\_\_(self, val=0, next=None):  
 self.val = val  
 self.next = next  
  
#to add two linked list and store result in third linked list  
class AddTwoNumbers:  
 def \_\_init\_\_(self):  
 self.ll1 = ListNode(0)  
 self.ll2 = ListNode(0)  
 self.ll3 = ListNode(0)  
  
#convert list to linked list  
 def createLinkedList(self, l1, l2):  
 self.h1 = self.ll1  
 for i in range(len(l1)):  
 self.ll1.val = l1[i]  
 if i != len(l1) - 1:  
 self.ll1.next = ListNode(0)  
 self.ll1 = self.ll1.next  
  
 self.h2 = self.ll2  
 for i in range(len(l2)):  
 self.ll2.val = l2[i]  
 if i != len(l1) - 1:  
 self.ll2.next = ListNode(0)  
 self.ll2 = self.ll2.next  
  
#adding each number from units place  
 def addTwoNumbers(self):  
 self.h3 = self.ll3  
 carry = 0  
 while self.h1 or self.h2 or carry:  
 if self.h1:  
 carry += self.h1.val  
 self.h1 = self.h1.next  
 if self.h2:  
 carry += self.h2.val  
 self.h2 = self.h2.next  
 self.ll3.val = carry % 10  
 carry = carry // 10  
 if self.h1 or self.h2 or carry:  
 self.ll3.next = ListNode(0)  
 self.ll3 = self.ll3.next  
 print("Addition Result:")  
 while self.h3:  
 print(self.h3.val)  
 self.h3 = self.h3.next  
  
  
#get input as list from user  
print("Enter values for linked list 1 with spaces")  
l1 = [int(x) for x in input().split()]  
print("Enter values for linked list 2 with spaces")  
l2 = [int(x) for x in input().split()]  
obj = AddTwoNumbers()  
obj.createLinkedList(l1, l2)  
obj.addTwoNumbers()

Leetcode:

# Definition for singly-linked list.

# class ListNode(object):

# def \_\_init\_\_(self, val=0, next=None):

# self.val = val

# self.next = next

class Solution(object):

def addTwoNumbers(self, l1, l2):

"""

:type l1: ListNode

:type l2: ListNode

:rtype: ListNode

"""

head = l3 = ListNode(0)

carry = 0

while l1 or l2 or carry:

if l1:

carry += l1.val

l1 = l1.next

if l2:

carry += l2.val

l2 = l2.next

l3.val = carry % 10

carry = carry // 10

if l1 or l2 or carry:

l3.next = ListNode(0)

l3 = l3.next

return head