Problem 5: Given the **heads** of two non-empty linked lists representing two non-negative integers. The digits are stored in **reverse order**, and each of their nodes contains a single digit. Add the two numbers and return the **sum** as a linked list.

```
class ListNode:
    def __init_(self, val=0, next=None):
        self.val = val
        self.next = next


def addTwoNumbers(l1, l2):
    dummy = ListNode()
    curr = dummy
    carry = 0


    p1, p2 = l1, l2
    while p1 or p2:
        x = p1.val if p1 else 0
        y = p2.val if p2 else 0
        _sum = x + y + carry


        carry = _sum // 10
        curr.next = ListNode(_sum % 10)


        curr = curr.next
        p1 = p1.next if p1 else None
        p2 = p2.next if p2 else None


    if carry:
        curr.next = ListNode(carry)


    return dummy.next
```

```
l1 = ListNode(2)

l1.next = ListNode(4)

l1.next.next = ListNode(3)


l2 = ListNode(5)

l2.next = ListNode(6)

l2.next.next = ListNode(4)



result = addTwoNumbers(l1, l2)



while result:
   print(result.val, end=" ")

   result = result.next
```