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 contains a single digit. Add the two numbers and return the sum as a linked list.

You may assume the two numbers do not contain any leading zero, except the number 0 itself.

Example 1:

Input: l1 = [2,4,3], l2 = [5,6,4] Output: [7,0,8] Explanation: 342 + 465 = 807.

Example 2:

Input: l1 = [0], l2 = [0] Output: [0]

Example 3:

Input: l1 = [9,9,9,9,9,9,9], l2 = [9,9,9,9] Output: [8,9,9,9,0,0,0,1]

Constraints:

The number of nodes in each linked list is in the range [1, 100]. 0 <= Node.val <= 9 It is guaranteed that the list represents a number that does not have leading zeros.

In [1]:

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

def addTwoNumbers(l1, l2):
    dummy = ListNode()  # Dummy head node of the result linked list
    curr = dummy  # Current node for constructing the result linked list
    carry = 0  # Carry-over value

    while l1 or l2:
        x = l1.val if l1 else 0
        y = l2.val if l2 else 0

        sum = x + y + carry
        carry = sum // 10

        curr.next = ListNode(sum % 10)
        curr = curr.next

        if l1:
            l1 = l1.next
        if l2:
            l2 = l2.next

    if carry > 0:
        curr.next = ListNode(carry)

    return dummy.next

# Example usage:
# Create linked list l1: 2-> 4-> 3
l1 = ListNode(2)
l1.next = ListNode(4)
l1.next.next = ListNode(3)

# Create linked list l2: 5-> 6-> 4
l2 = ListNode(5)
l2.next = ListNode(6)
l2.next.next = ListNode(4)

# Add the two numbers
result = addTwoNumbers(l1, l2)

# Print the result: 7-> 0-> 8
while result:
    print(result.val, end=" ")
    result = result.next
```

```
7 0 8
```

In [2]:

```
# Example usage:
# Create linked list l1: 0
l1 = ListNode()

# Create linked list l2: 0
l2 = ListNode()

# Add the two numbers
result = addTwoNumbers(l1, l2)

# Print the result: 0
while result:
    print(result.val, end=" ")
    result = result.next
```

```
0
```

In [3]:

```
# Example usage:
# Create linked list l1: 9-> 9-> 9-> 9-> 9-> 9-> 9
l1 = ListNode(9)
l1.next = ListNode(9)
l1.next.next = ListNode(9)
l1.next.next.next = ListNode(9)
l1.next.next.next.next = ListNode(9)
l1.next.next.next.next.next = ListNode(9)
l1.next.next.next.next.next.next = ListNode(9)

# Create linked list l2: 9-> 9-> 9-> 9
l2 = ListNode(9)
l2.next = ListNode(9)
l2.next.next = ListNode(9)
l2.next.next.next = ListNode(9)

# Add the two numbers
result = addTwoNumbers(l1, l2)

# Print the result: 8-> 9-> 9-> 9-> 0-> 0-> 0-> 1
while result:
    print(result.val, end=" ")
    result = result.next
```

```
8 9 9 9 0 0 0 1
```

In [ ]:

```

```