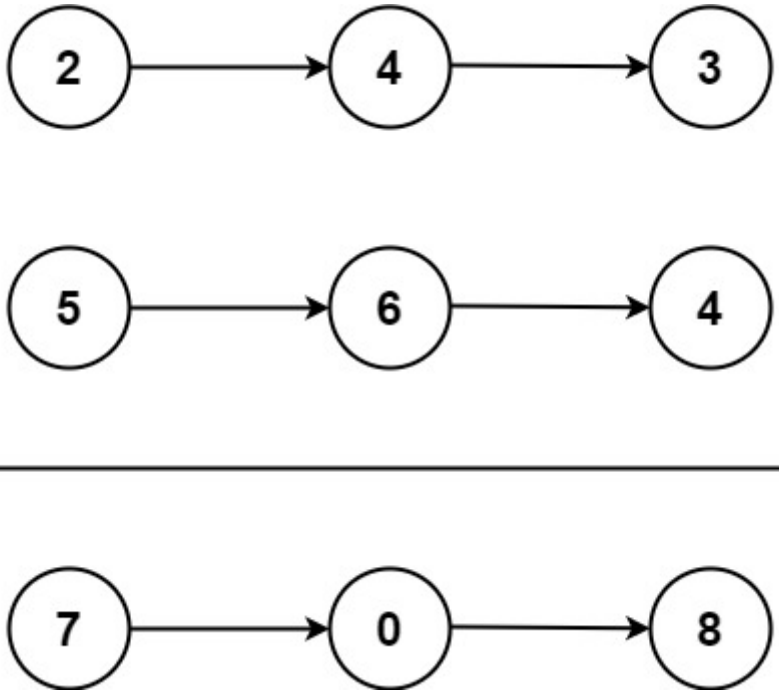
Example 1:




```
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.

Accepted 2,114,539 Submissions 5,774,416

```
/**
 * Definition
list.
 * type ListN
 *     Val in
 *     Next *
 * }
 */
func addTwoNu
l2 *ListNode)
    remainder
    origin :=
l2.Val, nil}
    l1, l2 =
    remainder
    origin.Va

    var pv *L
    pv = orig
    for l1 !=
remainder != 0
        nx :=
&ListNode{rem
        if l1
            n
            l
        }
        if l2
            n
            l
        }
        remai
        nx.Va
        pv.Ne
        pv = r
    }
    return or
}
```

Testcase Run Code Result

Accepted Runtime: 0

Your input [2,4,3] [5,6,4]

Output [7,0,8]

Expected [7,0,8]