```
def mySqrt(x):
    if x == 0:
        return 0

    left, right = 1, x
    while left <= right:
        mid = left + (right - left) // 2
        if mid * mid == x:
            return mid
        elif mid * mid < x:
            left = mid + 1
        else:
            right = mid - 1

    return right


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

def addTwoNumbers(l1, l2):
    dummy = ListNode(0)  # Dummy node to track the head of the result
    current = dummy  # Pointer to the current node in the result

    carry = 0  # Initialize the carry-over to 0

    while l1 or l2 or carry:
        # Calculate the sum of the current digits and the carry-over
        sum = (l1.val if l1 else 0) + (l2.val if l2 else 0) + carry

        carry = sum // 10  # Calculate the carry-over for the next place value
        digit = sum % 10  # Calculate the current digit for the result

        # Create a new node for the current digit and set it as the next node for the resu
        current.next = ListNode(digit)
        current = current.next

        # Move to the next nodes in the input linked lists if they exist
        if l1:
            l1 = l1.next
        if l2:
            l2 = l2.next

    return dummy.next  # Return the head of the result linked list
```