Problem 3: Given two singly linked lists that are sorted in increasing order of node values, merge two **sorted** linked lists and return them as a sorted list. The list should be made by splicing together the nodes of the first two lists.

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

def mergeTwoLists(l1, l2):
    dummy = ListNode(-1)
    current = dummy

    while l1 and l2:
        if l1.val <= l2.val:
            current.next = l1
            l1 = l1.next
        else:
            current.next = l2
            l2 = l2.next
        current = current.next

    current.next = l1 if l1 else l2

    return dummy.next


def printLinkedList(head):
    result = []
    while head:
        result.append(head.val)
        head = head.next
    return result
```

l1 = ListNode(3)

l1.next = ListNode(7)

l1.next.next = ListNode(10)

l2 = ListNode(1)

l2.next = ListNode(2)

l2.next.next = ListNode(5)

l2.next.next.next = ListNode(8)

l2.next.next.next.next = ListNode(10)

merged_list = mergeTwoLists(l1, l2)

print(printLinkedList(merged_list))