Program 41. Merge Two Sorted Lists You are given the heads of two sorted linked lists list1 and list2. Merge the two lists in a one sorted list. The list should be made by splicing together the nodes of the first two lists. Return the head of the merged linked list. Example 1: Input: list1 = [1,2,4], list2 = [1,3,4] Output: [1,1,2,3,4,4
PROGRAM:

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

def mergeTwoLists(list1, list2):
    dummy = ListNode()
    current = dummy

    while list1 and list2:
        if list1.val < list2.val:
            current.next, list1 = list1, list1.next
        else:
            current.next, list2 = list2, list2.next
        current = current.next

    current.next = list1 if list1 else list2
    return dummy.next

# Example usage
def printList(node):
    while node:
        print(node.val, end=" ")
        node = node.next
    print()

list1 = ListNode(1, ListNode(2, ListNode(4)))
list2 = ListNode(1, ListNode(3, ListNode(4)))

merged_head = mergeTwoLists(list1, list2)
printList(merged_head)  # Output: 1 1 2 3 4 4
```

Output::

```
"C:\Program Files\Python312\python.exe" "C:\Work Space\DAA COADS.PYTHON\program 41.py"
1 1 2 3 4 4

Process finished with exit code 0
```

Time complexity:
O(m+n)