1. 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:-

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

def mergeTwoLists(list1, list2):
    # Create a dummy node to act as the start of the merged list
    dummy = ListNode()
    current = dummy

    # Traverse both lists, adding the smaller node to the merged list
    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

    # If one of the lists is not empty, attach the remaining elements
    current.next = list1 if list1 else list2

    # Return the next node of dummy, which is the head of the merged list
    return dummy.next

# Helper function to create a linked list from a list
def createLinkedList(arr):
    if not arr:
        return None
    head = ListNode(arr[0])
    current = head
    for val in arr[1:]:
        current.next = ListNode(val)
        current = current.next
    return head

# Helper function to print a linked list
def printLinkedList(head):
    vals = []
    while head:
        vals.append(head.val)
        head = head.next
```

```
    print(vals)

# Example 2 usage: Both lists are empty
list1 = createLinkedList([])
list2 = createLinkedList([])
merged_list = mergeTwoLists(list1, list2)
printLinkedList(merged_list)  # Expected output: []
```

output:-

```python
[1, 1, 2, 3, 4, 4]
```

Time complexity:-O(n+m)