MERGE TWO SORTED LINKEDLIST

```java
package com.wipro;

public class MEREGESORTED_LIST {


    public static ListNode mergeTwoLists(ListNode l1, ListNode l2) {
        ListNode dummy = new ListNode(-1); // Dummy node to simplify edge cases
        ListNode current = dummy;

        while (l1 != null && l2 != null) {
            if (l1.value <= l2.value) {
                current.next = l1;
                l1 = l1.next;
            } else {
                current.next = l2;
                l2 = l2.next;
            }
            current = current.next;
        }

        // If one of the lists is not empty, append it to the result
        if (l1 != null) {
            current.next = l1;
        } else if (l2 != null) {
            current.next = l2;
        }

        return dummy.next; // Return the merged list, which starts after the dummy node
    }

    public static void main(String[] args) {
        // Create first sorted linked list: 1 -> 3 -> 5
        ListNode node1 = new ListNode(1);
        ListNode node2 = new ListNode(3);
        ListNode node3 = new ListNode(5);
        node1.next = node2;
        node2.next = node3;

        // Create second sorted linked list: 2 -> 4 -> 6
        ListNode node4 = new ListNode(2);
        ListNode node5 = new ListNode(4);
        ListNode node6 = new ListNode(6);
        node4.next = node5;
        node5.next = node6;

        // Merge the two sorted linked lists
        ListNode mergedList = mergeTwoLists(node1, node4);

        // Print the merged sorted linked list
        System.out.println("Merged sorted linked list:");
        printList(mergedList);
    }

    public static void printList(ListNode head) {
        ListNode current = head;
```

```
        while (current != null) {
            System.out.print(current.value + " ");
            current = current.next;
        }
        System.out.println();
    }
}
```