**Problem: 01**

**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]

**Example 2:**

**Input:** list1 = [], list2 = []

**Output:** []

**Example 3:**

**Input:** list1 = [], list2 = [0]

**Output:** [0]

**Constraints:**

* The number of nodes in both lists is in the range [0, 50].
* -100 <= Node.val <= 100
* Both list1 and list2 are sorted in **non-decreasing** order.

Code:

class Solution {

public ListNode mergeTwoLists(ListNode l1, ListNode l2) {

ListNode prehead = new ListNode(-1);

ListNode prev = prehead;

while (l1 != null && l2 != null) {

if (l1.val <= l2.val) {

prev.next = l1;

l1 = l1.next;

} else {

prev.next = l2;

l2 = l2.next;

}

prev = prev.next;

}

if(l1 == null){

prev.next = l2;

}

else if(l2 == null){

prev.next = l1;

}

return prehead.next;

}

}