To correctly implement the mergeTwoLists method that merges two sorted linked lists (list1 and list2) into one sorted list, you can use a **dummy node** and iterate through both lists comparing node values.

Here’s the **correct implementation**:

/\*\*

\* Definition for singly-linked list.

\* public class ListNode {

\* int val;

\* ListNode next;

\* ListNode() {}

\* ListNode(int val) { this.val = val; }

\* ListNode(int val, ListNode next) { this.val = val; this.next = next; }

\* }

\*/

class Solution {

public ListNode mergeTwoLists(ListNode list1, ListNode list2) {

// Create a dummy node to simplify logic

ListNode dummy = new ListNode(-1);

ListNode current = dummy;

// Traverse both lists and add the smaller value to the merged list

while (list1 != null && list2 != null) {

if (list1.val <= list2.val) {

current.next = list1;

list1 = list1.next;

} else {

current.next = list2;

list2 = list2.next;

}

current = current.next;

}

// Append the remaining part of list1 or list2

if (list1 != null) {

current.next = list1;

} else {

current.next = list2;

}

// Return the head of the merged list

return dummy.next;

}

}