**Task 7: Merging Two Sorted Linked Lists**

**You are provided with the heads of two sorted linked lists. The lists are sorted in ascending order. Create a merged linked list in ascending order from the two input lists without using any extra space (i.e., do not create any new nodes).**

**Sol:**

**Program:**

class ListNode {

int val;

ListNode next;

ListNode(int x) { val = x; }

}

public class MergeSortedLinkedLists {

public static ListNode mergeTwoLists(ListNode l1, ListNode l2) {

// Create a dummy node to serve as the start of the merged list

ListNode dummy = new ListNode(0);

// Use a pointer to build the new list

ListNode current = dummy;

// Traverse both lists

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

if (l1.val < l2.val) {

current.next = l1;

l1 = l1.next;

} else {

current.next = l2;

l2 = l2.next;

}

current = current.next;

}

// If there are remaining nodes in l1 or l2, append them

if (l1 != null) {

current.next = l1;

} else {

current.next = l2;

}

// The merged list starts from the node following the dummy node

return dummy.next;

}

public static void printList(ListNode node) {

while (node != null) {

System.out.print(node.val + " -> ");

node = node.next;

}

System.out.println("null");

}

public static void main(String[] args) {

// Example usage:

// List 1: 1 -> 3 -> 5

ListNode l1 = new ListNode(1);

l1.next = new ListNode(3);

l1.next.next = new ListNode(5);

// List 2: 2 -> 4 -> 6

ListNode l2 = new ListNode(2);

l2.next = new ListNode(4);

l2.next.next = new ListNode(6);

// Merge the lists

ListNode mergedList = mergeTwoLists(l1, l2);

// Print the merged list

printList(mergedList); // Expected: 1 -> 2 -> 3 -> 4 -> 5 -> 6 -> null

}

}