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).

1) **Steps for implementation:**
   1. Compare the values of the heads of the two lists.
   2. Take the smaller value and set it as the head of the merged list.
   3. Move the head pointer of the list whose node was selected for merging.
   4. Repeat steps 1-3 until one of the lists becomes empty.
   5. Append the remaining nodes of the non-empty list to the merged list.

JAVA Code:

```
class ListNode {
   int val;
   ListNode next;
   ListNode(int val) { this.val = val; }
}
public class MergeSortedLinkedLists {
   public static ListNode mergeLists(ListNode l1, ListNode l2) {
      ListNode dummy = new ListNode(-1);
      ListNode current = dummy;

      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;
```

```
        }
      current.next = (l1 != null) ? l1 : l2;
      return dummy.next;
   }
   public static void main(String[] args)
       {
      ListNode l1 = new ListNode(1);
      l1.next = new ListNode(3);
      l1.next.next = new ListNode(5);
      ListNode l2 = new ListNode(2);
      l2.next = new ListNode(4);
      l2.next.next = new ListNode(6);
      ListNode mergedList = mergeLists(l1, l2);
      printList(mergedList);
   }

   public static void printList(ListNode head)
       {
      ListNode current = head;
      while (current != null) {
         System.out.print(current.val + " ");
         current = current.next;
      }
   }
}
```

Explanation of the Code

1. **mergeLists Method:**
   - Initializes a dummy node to start the merged list.
   - Iterates through the lists while both lists are not empty.
   - Compares the values of the current nodes of the two lists.
   - Appends the smaller value node to the merged list and moves the pointer of that list.
   - Continues this process until one of the lists becomes empty.
   - Appends the remaining nodes of the non-empty list to the merged list.
2. **printList Method**:
   - Helper function to print the values of the merged list.
3. **Main Method:**
   - Creates two sorted linked lists l1 and l2.
   - Calls the mergeLists function to merge the two lists.
   - Prints the merged list.