

```

/*
将两个有序链表合并为一个新的有序链表并返回。新链表是通过拼接给定的两个链表的所有节点组成的。
示例：
输入：1->2->4, 1->3->4
输出：1->1->2->3->4->4
来源：力扣（LeetCode）
链接：https://leetcode-cn.com/problems/merge-two-sorted-lists
著作权归领扣网络所有。商业转载请联系官方授权，非商业转载请注明出处。
*/

```

分析：

- 常规方法：双指针依次遍历两个链表的所有节点，使用辅助`head`节点作为头节点将遍历结果串起来，然后返回`head.next`即可。

方法1：C_双指针逐一遍历

```

/**
 * Definition for singly-linked list.
 * struct ListNode {
 *     int val;
 *     struct ListNode *next;
 * };
 */
struct ListNode* mergeTwoLists( struct ListNode* l1 ,
                                struct ListNode* l2
                                )
{
    struct ListNode head ;
    struct ListNode* tail_node = NULL ;

    head.next = NULL;
    tail_node = &head;
    while(1)
    {
        if((l1!=NULL)&&(l2!=NULL))
        {
            if(l1->val <= l2->val)
            {
                tail_node->next = l1 ;
                tail_node = l1 ;
                l1 = l1->next ;
                tail_node->next = NULL ;
            }
            else
            {
                tail_node->next = l2 ;
                tail_node = l2 ;
                l2 = l2->next ;
                tail_node->next = NULL ;
            }
        }
    }
}

```

```
    else if(l1!=NULL)
    {
        tail_node->next    =    l1        ;
        tail_node          =    l1        ;
        l1                  =    l1->next  ;
        tail_node->next    =    NULL      ;
    }
    else if(l2!=NULL)
    {
        tail_node->next    =    l2        ;
        tail_node          =    l2        ;
        l2                  =    l2->next  ;
        tail_node->next    =    NULL      ;
    }
    else
    {
        break;
    }
}
return head.next;
}
/*
执行结果:
通过
显示详情
执行用时 :8 ms, 在所有 C 提交中击败了68.29% 的用户
内存消耗 :7.5 MB, 在所有 C 提交中击败了77.42%的用户
*/
```