Homework：链表

2.将两个非递减的有序链表合并为一个非递增的有序链表。要求结果链表仍使用原来两个链表的存储空间，不另外占用其他的存储空间。表中允许有重复的数据。

```
class ListNode {
public:
    ListNode* merge_list(ListNode* l1, ListNode* l2)
    {
        ListNode* p = nullptr;
        //非递增，无头节点，先找大
        if (l1->val <= l2->val)
        {
            p = l2;
            l2 = l2->next;
        }
        else
        {
            p = l1;
            l1 = l1->next;
        }
        while (l1 && l2)
        {
            if (l1->val <= l2->val)
            {
                p->next = l2;
                l2 = l2->next;
            }
            else
            {
                p->next = l1;
                l1 = l1->next;
            }
        }
        if (l1)
        {
            p->next = l1;
        }
        else
        {
            p->next = l2;
        }
        return p;
    }

private:
    int val;
    ListNode* next;
};
```

4.已知两个链表A和B分别表示两个集合，其素递增排列。请设计算法求出两个集合A和B的差集(即仅由在A中出现而不在 B 中出现的素所构成的集合)，并以同样的形式存储同时返回该集合的元素个数。

```
class ListNode {
public:
    ListNode* diffrence_list(ListNode* l1, ListNode* l2)
    {
        //l1,l2是带头节点的链表
        ListNode* head = l1;
        ListNode* pre = l1;
        l1 = l1->next;
        l2 = l2->next;
        while (l1 && l2)
        {
            if (l1->val == l2->val)
            {
                pre->next = l1 ->next;
                delete l1;
                l1 = pre->next;
            }
            else if (l1->val > l2->val)
            {
                l2 = l2->next;
            }
            else
            {
                pre = l1;
                l1 = l1->next;
            }
        }
        return head;
    }

private:
    int val;
    ListNode* next;
};
```

6.设计一个算法，通过一趟遍历确定长度为 n的单链表中值最大的结点。

```
class ListNode {
public:
    ListNode* max_node(ListNode* l1)
    {
        ListNode* max = l1;
        while (l1)
        {
            if (l1->val > max->val)
            {
                max = l1;
            }
            l1 = l1->next;
        }
        return max;
    }

private:
    int val;
    ListNode* next;
};
```