1、 某非空单链表 L 中的所有元素为整数, 设计一个算法将所有小于零的结点移到所有大于等于零的结点的前面。

```
typedef int ElemType;
typedef struct LNode {
    ElemType data;
    struct LNode *next;
}LNode,*LinkList;


//将所有小于零的结点移到所有大于等于零的结点的前面
Status ListMove_L(LinkList &L) {
    LinkList pre,p;
    if (!L->next) return ERROR;
    pre = L->next; p = pre->next;
    while (p){
        if (p->data<0){
            pre->next = p->next;
            p->next = L->next;
            L->next = p;
            p = pre->next;
        }
        else{
            pre = p;
            p = p->next;
        }
    }
     return OK;
} // ListMove_L
```

2、 有一个递增单链表 L(允许出现值域重复的结点),设计一个算法删除值域重复的结点。

```
typedef int ElemType;
typedef struct LNode {
    ElemType data;
    struct LNode *next;
}LNode,*LinkList;


//删除递增单链表中值域重复的结点
Status ListUnrepeat_L(LinkList &L) {
    LinkList p, q;
    if (!L->next) return ERROR;
    p = L->next;
    while (p->next){
        if (p->next->data==p->data) {
            q = p->next;
            p->next = q->next;
            free(q);
        }
        else p = p->next;
    }
    return OK;
} //ListUnrepeat_L
```