作业二参考答案

2.8.1

```
// 2.8.1
Status DelDup(Linklist L)
// 删除无序链表中重复的元素，带头结点
    for (p=L->next; p; p=p->next) {
    // 遍历链表
        for (q=p->next,r=p; q; r=q,q=q->next) {
        // 判断是否有重复元素
            if (p->data == q->data) {
                r->next = q->next;
                free(q);
                q = r;
            }
        }
    }
    return OK;
}// DelDup
```

2.8.2

```
// 2.8.2
Status DelDup_Order(LinkList L)
// 删除非递减链表的中的重复元素，带头结点
{
    for(p=L->next; p->next; p=p->next){
    // 遍历链表，比较相邻量结点是否相等
        while(p->data == p->next->data){
            q=p->next; p->next=q->next;
            free(q);
        }
    }
    return OK;
}// DelDup_Order
```

2.10

```
// 2.10
Status DelPrior(LNode *s)
// 删除循环链表s结点的前驱
{
    p = s;
    // 找到s结点的前驱的前驱p
    while(p->next->next != s) p=p->next;
    free(p->next);
    p->next = s;
    return OK;
}// DelPrior
```

2.12

```
// 2.12
Status PartOddEven(SqList L)
// 将顺序表中的奇数放在左侧，偶数放在右侧
{
    temp = L.elem[0];
    i=0; j=L.length-1; dir=0;
```

```
    while(i<j){
        if(dir==0){ // 从后向前找到第一个奇数
            if(L.elem[j]%2==1){ // 奇数
                L.elem[i] = L.elem[j];
                i++; dir=1;
            }
            else j--;
        }
        else{ // 从前向后找第一个偶数
            if(L.elem[i]%2==0){ // 偶数
                L.elem[j] = L.elem[i];
                j--; dir=0;
            }
            else i++;
        }
    }// while
    L.elem[i] = temp;
}// PartOddEven
```