第1关：数据元素插入

**任务描述**

**本关任务**：已知顺序表L中的数据元素递增有序，数据元素类型为int。相关定义如下： #define LIST\_INIT\_SIZE 20 #define LISTINCREMENT 10 typedef int ElemType; typedef struct { ElemType \*elem; int length; int listsize; } SqList; 试写一算法，将x插入到顺序表的适当位置上，以保持该表的有序性。 **函数原型**：int insert(SqList &L,ElemType x);

**相关知识**

为了完成本关任务，你需要掌握：1.线性表的物理结构，2.线性表的运算。

**编程要求**

根据提示，在右侧编辑器补充代码，写出函数 int insert(SqList &L,ElemType x) 的定义。

**测试说明**

平台会自动读取输入数据，对你编写的代码进行测试，并输出结果。

测试输入：0 1 预期输出： 1

测试输入：1 3 5 7 9 0 6 预期输出： 1 3 5 6 7 9

第2关：双向链表

**任务描述**

**本关任务**：设以带头结点的双向循环链表表示线性表L=（a1，a2，…，an），试编写一个时间复杂度为O(n)的算法，将L改变成L=（a1，a3，…，an，…，a4，a2）。 相关定义如下： typedef int ElemType; typedef struct Dnode { ElemType data; struct Dnode \* prior, \* next; } \* DuLinkList; **函数原型**：void adjust(DuLinkList L);

**相关知识**

为了完成本关任务，你需要掌握：1.线性表的物理结构，2.双向循环链表的插入与删除。

**编程要求**

根据提示，在右侧编辑器补充代码，写出函数 void adjust(DuLinkList L) 的定义。

**测试说明**

平台会自动读取输入数据，对你编写的代码进行测试，并输出结果。

测试输入：6 1 2 3 4 5 6 预期输出：1 3 5 6 4 2

测试输入：11  
2 11 3 5 7 6 12 13 8 9 20 预期输出：2 3 7 12 8 20 9 13 6 5 11

第3关：多顺序表操作

**任务描述**

**本关任务**：已知A、B和C为3个递增有序的线性表，现要求对A表做如下操作，删除那些既在B中出现，也在C中出现的元素。以顺序表作为线性表的物理结构，编写实现上述操作的算法。 **函数原型**：void TriSqList(SqList &A,SqList B,SqList C)；

**相关知识**

为了完成本关任务，你需要掌握：1.线性表，2.顺序表。 相关定义： typedef int ElemType; typedef struct { ElemType \*elem; int length; int listsize; } SqList;

**编程要求**

根据提示，在右侧编辑器补充代码，完成函数TriSqList(SqList &A,SqList B,SqList C)。 要求算法效率尽可能高。

**测试说明**

平台会自动读取输入三个线性表的数据，对你编写的代码进行测试，并输出结果。

测试输入：1 3 5 7 9 10 11 12 20 0

3 9 10 11 30 0

3 6 7 9 11 13 50 0 预期输出：1 5 7 10 12 20

第4关：多单链表操作

**任务描述**

**本关任务**：已知A、B和C为3个递增有序的线性表，现要求对A表做如下操作，删除那些既在B中出现，也在C中出现的元素。以带表头结点的单链表作为线性表的物理结构，编写实现上述操作的算法。 **函数原型**：void TriLinkList(LinkList A,LinkList B,LinkList C)；

**相关知识**

为了完成本关任务，你需要掌握：1.线性表，2.顺序表。 相关定义： typedef int ElemType; typedef struct node { ElemType data; struct node \*next; } NODE, \*LinkList;

**编程要求**

根据提示，在右侧编辑器补充代码，完成函数TriLinkList(LinkList A,LinkList B,LinkList C)。要求算法效率尽可能高。

**测试说明**

平台会自动读取输入三个线性表的数据，对你编写的代码进行测试，并输出结果。

测试输入：1 3 5 7 9 10 11 12 20 0 3 9 10 11 30 0 3 6 7 9 11 13 50 0 预期输出：1 5 7 10 12 20