

顺序表实验题

【实验目的】

1. 通过本实验，掌握线性表的顺序存储结构的定义及 C 语言的实现；
2. 熟练掌握线性表的基本操作在顺序存储结构上的实现。

【实验要求】

用 C 语言的结构体类型定义顺序表如下：

```
#define INIT_LIST_SIZE 6      //初始分配量
#define LISTINCREMENT 3      //分配增量
typedef int ElemType;         //顺序表中存放的是整数
typedef struct {
    ElemType *elem;           //存储空间基址
    int length;               //当前长度
    int listsize;             //当前分配的存储容量
}SqList;
```

【实验内容】

1.题目内容：

编程实现顺序表的下列基本操作函数。

(1) void InitList(SqList &L) //建立空的顺序表

(2) void ListInsert(SqList &L, int i, ElemType e)
//在顺序表中第 i 个位置插入元素函数 e

(3) void ListDelete(SqList&L, int i, ElemType &e)
//删除顺序表 L 的第 i 个数据元素，并用 e 返回其值。

(4) void PrintList(SqList L) // 输出顺序表

(5) int Locate(SqList L, ElemType e) /*若顺序表 L 中存在数据元素 e，则返回 e 在顺序表 L 中第一次出现的位序;否则返回 0.*/

输入格式：

四行数据，第一行的整数表示要建立的顺序表的元素个数，第二行整数表示依次输入的数据，两个整数之间以空格分隔，第三行的整数表示要删除数的位置，第四行表示要查找的数。

输出格式：

三行数据，第一行表示程序执行后顺序表的数据元素(依次从表头至表尾),第二行整数表示已删除的数，第三行表示要查找的数的位置或者没有找到。

输入样例：

5
4 3 6 7 9

3

4

输出样例：

4 3 7 9

6

1

输入样例：

7
2 1 5 8 7 6 9

1

4

输出样例：

1 5 8 7 6 9

2

Not find

2.题目内容：

已知顺序表中的元素依值递增有序排列，要求将 x 插入到顺序表的适当位置上，使得插入操作后的顺序表仍保持有序性。

输入格式：

三行数据，第一行的整数表示要建立的顺序表的元素个数，第二行整数表示依次输入的数据，两个整数之间以空格分隔，第三行的整数表示要插入的数 x 。

输出格式：

一行整数，从表头开始依次显示插入操作后顺序表的数据元素。

输入样例：

5
2 5 8 11 20

10

输出样例：

2 5 8 10 11 20

输入样例：

5
2 5 8 11 20

1

输出样例：

1 2 5 8 11 20

输入样例：

5
2 5 8 11 20

25

输出样例：

2 5 8 11 20 25

3.题目内容：

已知顺序表中的元素依值递增有序排列，要求删除表中所有值相同的多余元素（使得操作后的顺序表中所有元素的值均不相同）

输入样例：

8

2 5 5 8 11 20 20 20

输出样例：

2 5 8 11 20

4.题目内容:

假设以两个元素依值递增有序排列的顺序表 **A** 和 **B** 分别表示两个集合（同一表中的元素值各不相同），现要求另辟空间构成一个顺序表 **C**，其元素为 **A** 和 **B** 元素的交集，且表 **C** 中的元素也是依值递增有序排列。

输入样例：

5

2 5 8 11 20

6

3 8 9 11 15 20

输出样例：

8 11 20

5.题目内容:

设 $A=(a_1,...,a_m)$ 和 $B=(b_1,...,b_n)$ 均为顺序表， A' 和 B' 分别为 **A** 和 **B** 中除去最大共同前缀后的子表，若 $A'=B'$ =空表，则 $A=B$; 若 A' =空表，而 $B'!=$ 空表，或者两者都不为空表，且 A' 的首元小于 B' 的首元，则 $A<B$; 否则 $A > B$. 试编程，求出 **A,B** 的大小。

输入样例：

5

2 5 8 11 20

4

2 9 11 15

输出样例：

$A<B$

输入样例：

5

2 5 8 11 20

4

2 5 8 11

输出样例：

$A>B$

输入样例：

5

2 5 8 11 20

5

2 5 8 11 20

输出样例：

$A=B$