实验一 顺序表实验

1.1实验目的

1. 熟练掌握线性表的顺序存储结构。
2. 熟练掌握顺序表的有关算法设计和实现。
3. 根据具体问题的需要，设计出合理的表示数据元素的顺序结构，并设计相关算法。

1.2 实验要求

1. 顺序表结构和运算定义，算法的实现以库文件方式实现，不得在测试主程序中直接实现；比如存储、算法实现放入文件：seqList.h
2. 程序有适当的注释，程序的书写要采用缩进格式。
3. 实验程序有较好可读性，各运算和变量的命名直观易懂，符合软件工程要求；
4. 程序要具有一定的健壮性，即当输入数据非法时，程序也能适当地做出反应，如插入删除时指定的位置不对等等。
5. 程序要做到界面友好，在程序运行时用户可以根据相应的提示信息进行操作。
6. 程序运行、测试正确；
7. 根据实验报告模板详细书写实验报告。

1.3 实验任务

编写算法实现下列问题的求解。

1. 在一个递增有序的顺序表L中插入一个值为x的元素，并保持其递增有序特性。

实验测试数据基本要求：

顺序表元素为 （10,20,30,40,50,60,70,80,90,100）,

x分别为25，85，110和8

1. 将顺序表Ｌ中的奇数项和偶数项结点分解开（元素值为奇数、偶数），分别放入新的顺序表中，然后原表和新表元素同时输出到屏幕上，以便对照求解结果。

实验测试数据基本要求：

第一组数据：顺序表元素为 （1,2,3,4,5,6,7,8,9,10,20,30,40,50,60）

第二组数据：顺序表元素为 （10,20,30,40,50,60,70,80,90,100）

1. 求两个递增有序顺序表L1和L2中的公共元素，放入新的顺序表L3中。

实验测试数据基本要求：

第一组

第一个顺序表元素为 （1，3，6，10，15，16，17，18，19，20）

第二个顺序表元素为 （1，2，3，4，5，6，7，8，9，10，18，20，30）

第二组

第一个顺序表元素为 （1，3，6，10，15，16，17，18，19，20）

第二个顺序表元素为 （2，4，5，7，8，9，12，22）

第三组

第一个顺序表元素为 （）

第二个顺序表元素为 （1，2，3，4，5，6，7，8，9，10）

1. 删除递增有序顺序表中的重复元素，并统计移动元素次数，要求时间性能最好。

实验测试数据基本要求：

第一组数据：顺序表元素为 （1,2,3,4,5,6,7,8,9）

第二组数据：顺序表元素为 （1,1,2,2,2,3,4,5,5,5,6,6,7,7,8,8,9）

第三组数据：顺序表元素为 （1,2,3,4,5,5,6,7,8,8,9,9,9,9,9）

1.4\* 顺序表扩展实验

（非必做内容，有兴趣的同学选做）

1.（递增有序）顺序表表示集合A、B，实现：

C=A∪B，C=A∩B，C=A-B

A=A∪B，A=A∩B，A=A-B

2. (2011)（15 分）一个长度为L（L≥1）的升序序列S，处在第 个位置的数称为S 的中位数。例如，若序列S1=(11, 13, 15, 17, 19)，则S1 的中位数是15。两个序列的中位数是含它们所有元素的升序序列的中位数。例如，若S2=(2, 4, 6, 8, 20)，则S1 和S2 的中位数是11。



现有两个等长升序序列A 和B，试设计一个在时间和空间两方面都尽可能高效的算法，找出两个序列A 和B 的中位数。要求：

（1）给出算法的基本设计思想。

（2）根据设计思想，采用C 或C++语言描述算法，关键之处给出注释。

（3）说明你所设计算法的时间复杂度和空间复杂度。