**实 验 报 告**

|  |  |  |
| --- | --- | --- |
| **课程名称**：数据结构 | **班级：**软工21101 | **实验成绩**： |
| **实验名称**：顺序表的实现 | **学号**：2105006207 | **批阅教师签字：** |
| **实验编号**：实验一 | **姓名**：方福涛 | **实验日期：**2022 年 9 月 9 日 |
| **指导教师**：董傲霜 | **组号**： | **实验时间**：8 时20 分－ 10 时00 分 |

1. **实验设计思想**

1.设计思想：利用数组来实现顺序存储、推算线性表的存储位置、注意线性表和数组 的区别、线性表获取元素操作、顺序存储结构的插入以及删除。

2.线性表顺序存储结构实现的特点：逻辑关系上相邻的两个元素在物理位置上也相邻，因此可以随机存取表中的任意一个元素，它的存储位置可用一个简单、直观的公式来表示。

**二、程序说明**

1. 在建立顺序表时，除了预先申请内存空间，还需要实时记录顺序表的长度和顺序表本身申请的内存大小，便于后期对顺序表中的数据元素进行调取。
2. typedef struct list{

ElemType data[N];//定义 ElemType为整型数组有100个元素

int len;//数组中实际元素的个数

}list;//初始化

插入：找到要插入的位置，将后续数据元素整体向后移动一个位置，最后直接在腾 出来的位置上插入数据元素。

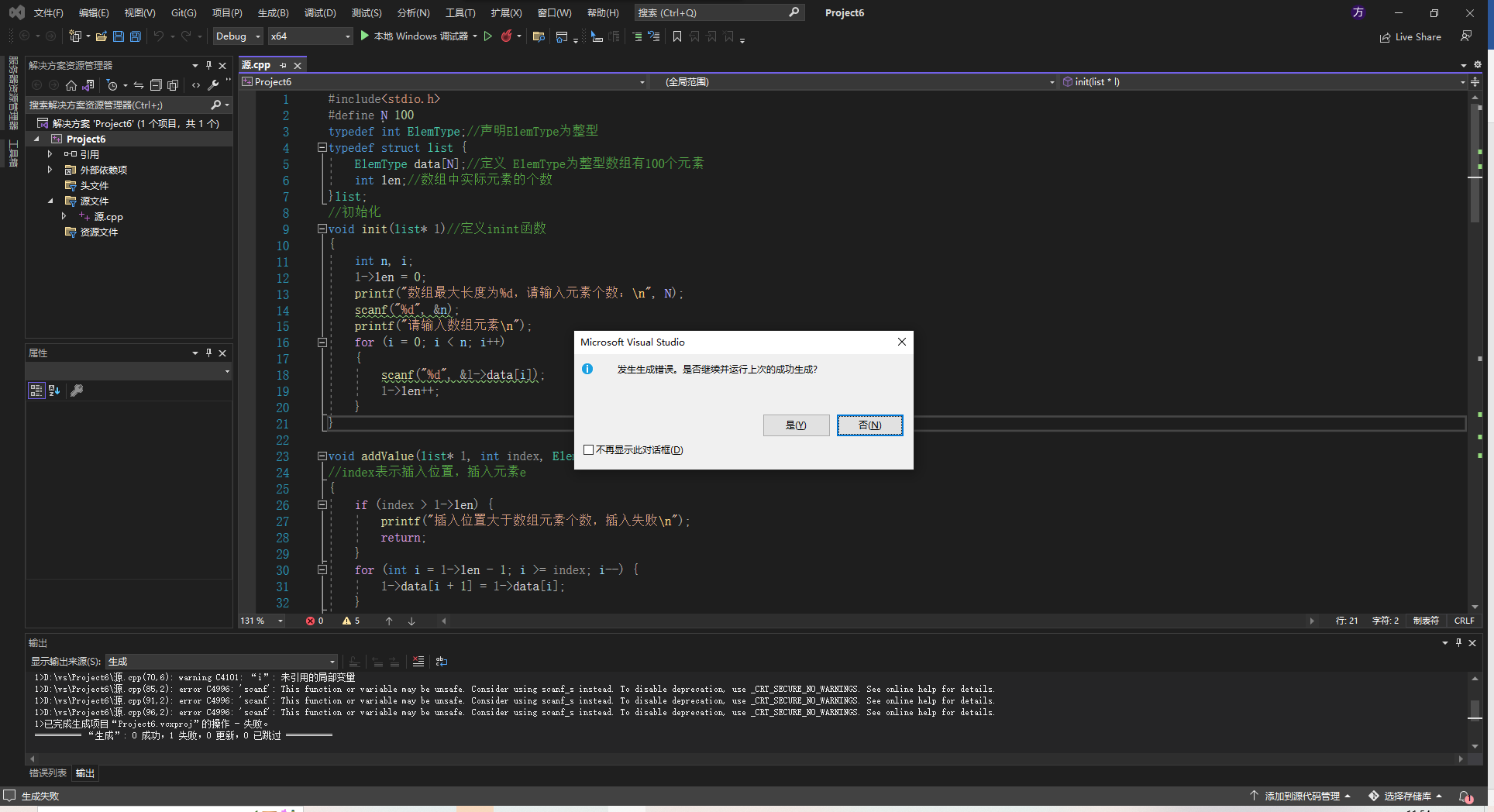
删除：在数组中删除元素时，只需将该元素所在位置后的所有数据元素整体前移 1 个位置即可。

更改：调用查找算法找到该数据元素的位置，直接在该位置上更改。

**三、实验环境**

**Windows、DEV-C++ 5.11**

**四、实验过程分析**



**在编码过程中，用vs2022发现报错，与c++中的一些语法使用不兼容，故此换成 DEV-C++，运行成功。**

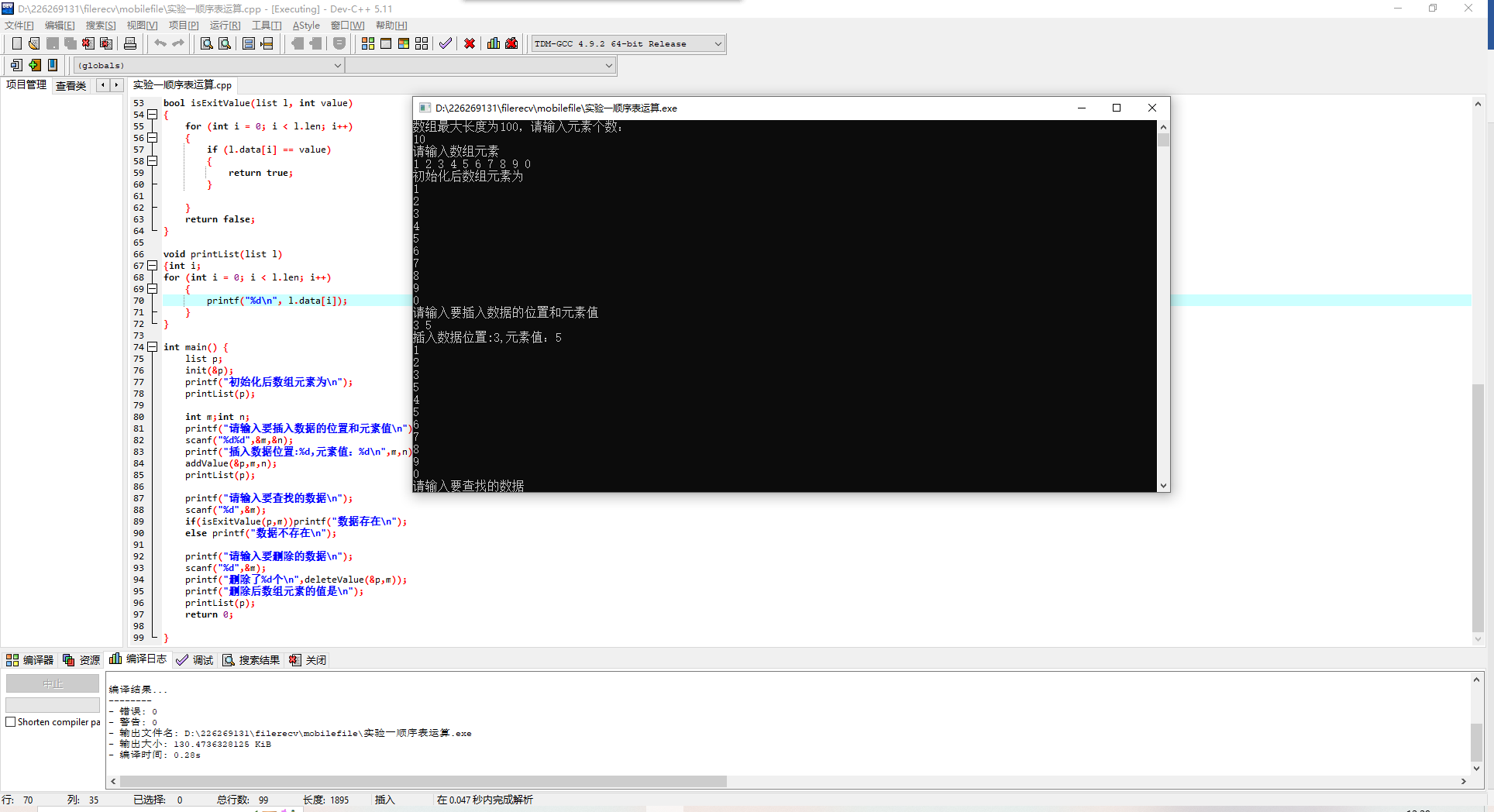
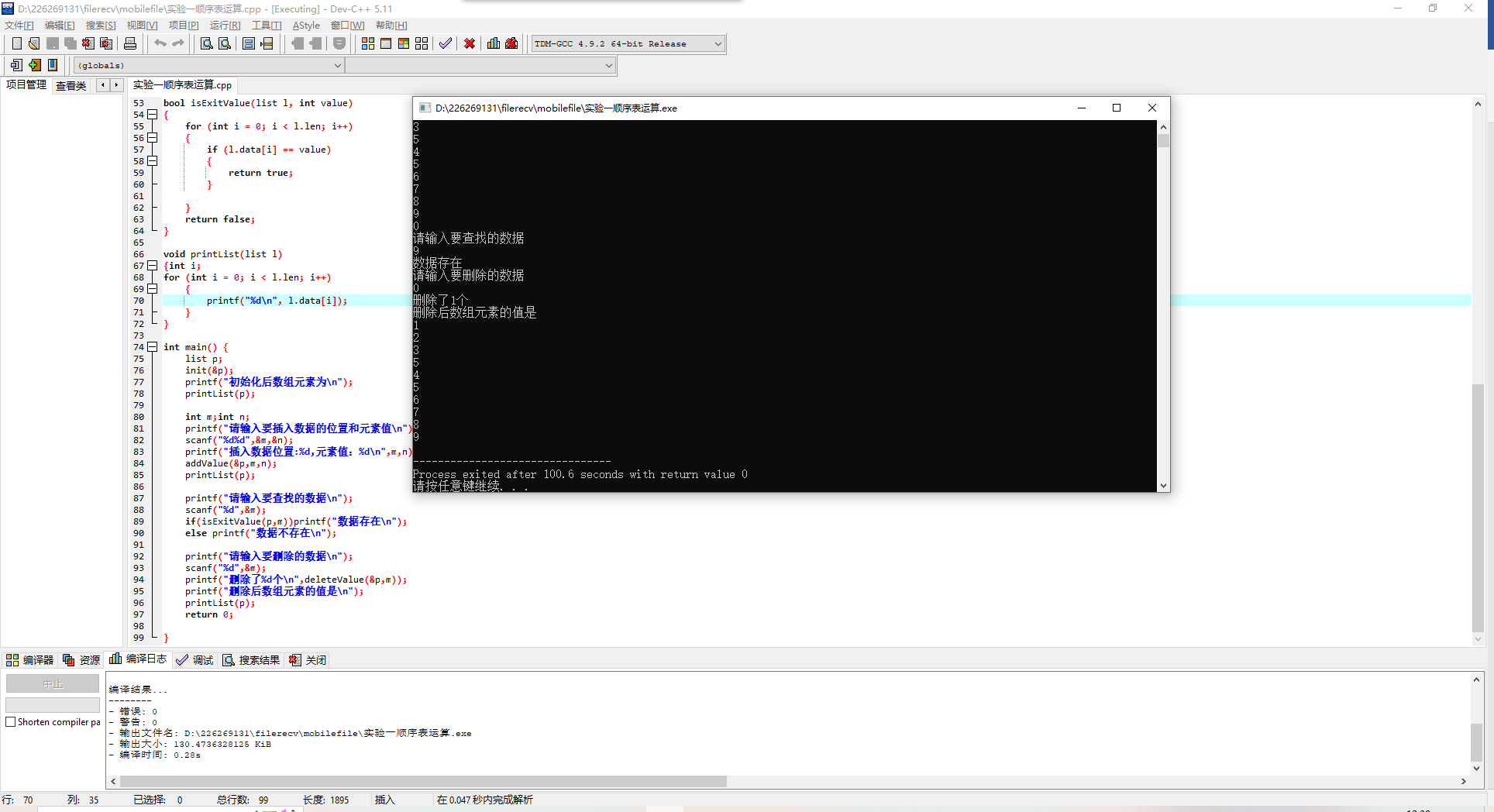
**在实验中也犯过很多低级错误，例如分析很久的报错，最后发现是单词拼写问题，以及空格，函数的定义都有相关的问题。**

1. **实验结果总结**

插入：在表头插入、在表的中间某个位置插入、 直接尾随顺序表，作为表的最后 一个元素。

删除：前移的过程中被删除元素被后一个元素覆盖掉，间接实现了删除操作。

更改：调用查找算法找到该数据元素的位置，直接在该位置上更改。

2、 

**六、附录**

1. 意见和建议（没有可不写）。

2. 思考题：

回答以下问题：

1. 为什么说线性表的顺序存储结构是线性表的紧凑存储结构？

逻辑上具有线性关系的数据按照前后的次序全部存储在一整块连 续的存空间中，之间不存在空隙。

1. 线性表的顺序存储结构适合用于什么特征的线性表的存储实现？？

线性表的存储结构，当需要快速查找如使用折半查找等算法时，需要采用顺序结构，可以大大提高查找的效率，及较稳定的线性表选择顺序存储。

**七、打分表**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 项目 | 考核点 | 分数 | **得分** | 备注 |
| 程序 | 逻辑是否正确  程序可读性  创新点  是否符合OOD设计原则 | 50 | **0** |  |
| 报告完整性 | 实验过程阐述是否完整  测试数据设计是否合理  运行结果是否正确 | 40 | **0** |  |
| 调试问题及解决方法 | 是否对调试过程问题进行阐述 | 5 | **0** |  |
| 思考题目 | 回答是否正确 | 5 | **0** |  |
| 合计 |  | 100 | **0** |  |