

湖 北 大 学

网络空间安全学院

2024 ~ 2025 学年度 第 1 学期

学 生 实 验 报 告 手 册

课 程 名 称: 数据结构

任 课 老 师: 李胜华

专 业 年 级: 信息安全 2303、2304

组号	第 2 组		
序号/学号/姓 名	36	202331120011063	付志城
	12	202331120011017	骆骏辉
	5	202331112052037	黄梓涵
	41	202331120011080	李祥
	59	202331120011110	王赫权
	19	202331120011029	敖元坤

学生实验守则

- 1、本课程所有实验作业**严禁抄袭雷同**。
- 2、实验作业的相关数据要实事求是，理论及结果分析合理、实验过程真实。
- 3、实验作业按时上交。

实 验 报 告 单

实验名称:	线性表的实现		
报告日期:	2024.9.12——2024.9.26	成 绩:	

【实验内容】

1. 实现顺序表/链表的各种基本运算
2. 求集合的运算
3. 力扣练习

【实验目的】掌握线性表的两种实现方式及相关算法，领会顺序表/单链表的应用

【实验过程及结果分析】

一、项目清单

1. 文件



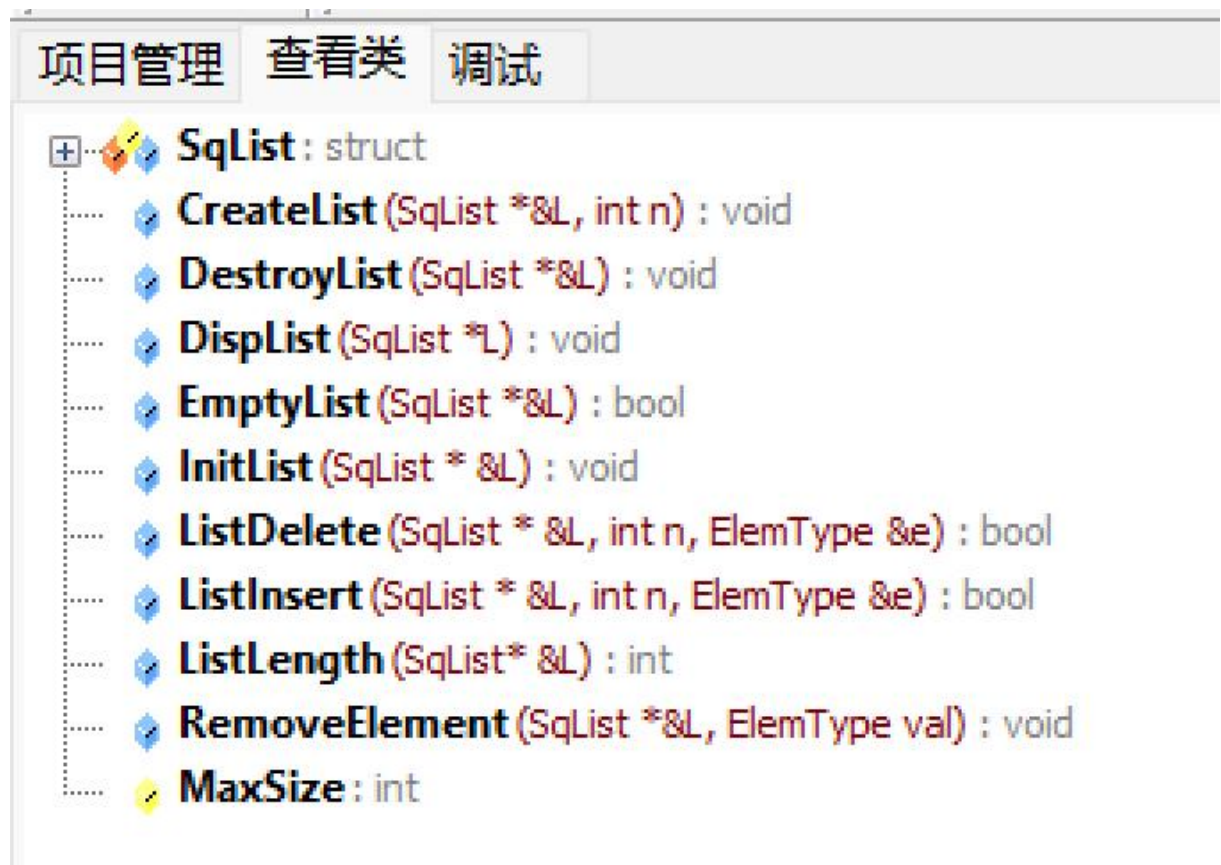
2. 数据存储

```

const int MaxSize=100;
typedef int ElemType;
typedef struct
{
    ElemType date[MaxSize];
    int len;
}SqList;

```

3. 功能函数



二、主要功能测试截屏（含主函数文件截屏）

付志城：

1. 创建

```
int main()
{
    int a[10] = {0, 1, 2, 3, 4, 56, 1, 2, 5, 6, };

    Sqlist* L;

    Initlist(L);
    cout << Emptylist(L) << endl;
    Creatlist(L, a, 8);
    cout << Emptylist(L) << endl;
    //cout<<"56的位置" << LocateElem(L, 56);
    return 0;
}
```

2.

(int)1
联机搜索

Microsoft Visual S
1
0
E:\c++\数据结构
要在调试停止时自
按任意键关闭此窗

2 按元素值查找

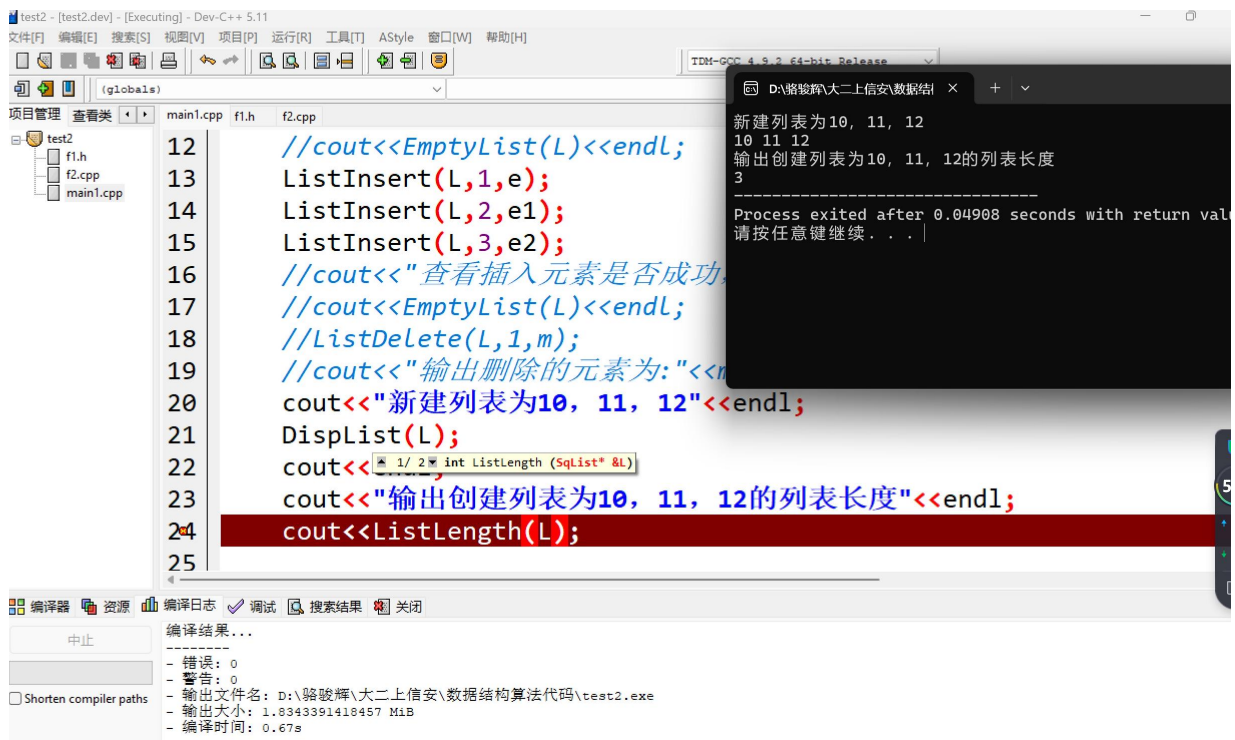
```
#include<iostream>
using namespace std;
#include"2.h"
int main()
{
    int a[10] = {0, 1, 2, 3, 4, 56, 1, 2, 5, 6, };

    Sqlist* L;
    Initlist(L);
    cout << Emptylist(L) << endl;
    Creatlist(L, a, 8);
    cout<<"56的位置" << LocateElem(L, 56);
    return 0;
}
```

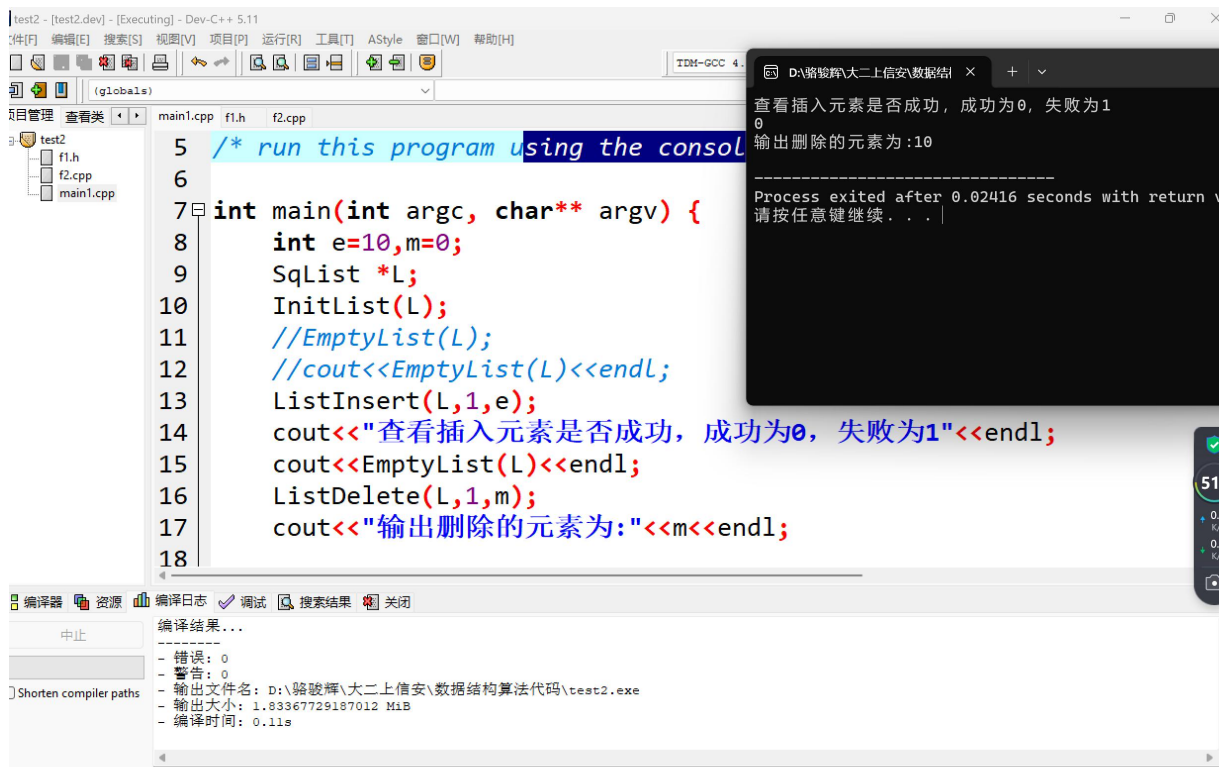
```
1
2 56的位置6
3 E:\c++\数据结构\x64\Debu
4 要在调试停止时自动关闭控
5 按任意键关闭此窗口...
```

骆骏辉:

1. 求长度



2. 删除



王赫权:

1. 判断是否为空

```
int main()
{
    SqList* L1 = NULL;
    Input(L1);
    Output(L1);
    IsEmptySqList(L1);
    DestroyList(L1);
}
```

Microsoft Visual Studio 调试

请输入数据的个数: 3
请输入数据来创建一个顺序表: 1 2 3
1 2 3
顺序表不为空

C:\Users\whq13\source\repos\C++\x64\Debug\C++.exe (进程 21388)已退出, 代码为 0 (0x0)。
按任意键关闭此窗口。 . . .

2. 逆置

```
int main()
{
    SqList* L1 = NULL;
    Input(L1);
    Output(L1);
    ReverseSqList(L1);
    Output(L1);
    DestroyList(L1);
}
```

Microsoft Visual Studio 调试

请输入数据的个数: 3
请输入数据来创建一个顺序表: 1 2 3
1 2 3
3 2 1

C:\Users\whq13\source\repos\C++\x64\Debug\C++.exe (进程 20792)已退出, 代码为 0 (0x0)。
按任意键关闭此窗口。 . . .

李祥:

1. 输出

```
新文件1.cpp
35     cout << endl;
36 }
37
38 void CreateList(SqList
39 {
40     L = (SqList *)mall
41
42
43
44
45     int i = 0, k = 0;
46
47     while (i < n) {
48
49         L->data[k] = a
50         k++;
51         i++;
52     }
53     L->length = k;
54 }
55
56 int main() {
57     SqList *A;
58     int a[5] = {1, 2, 3, 4, 5};
59     // L = (SqList *)malloc(sizeof(SqList));
60     CreateList(A, a, 4);
61     // Listinsert(A, 3, 7);
62     DispList(A);
63 }
```

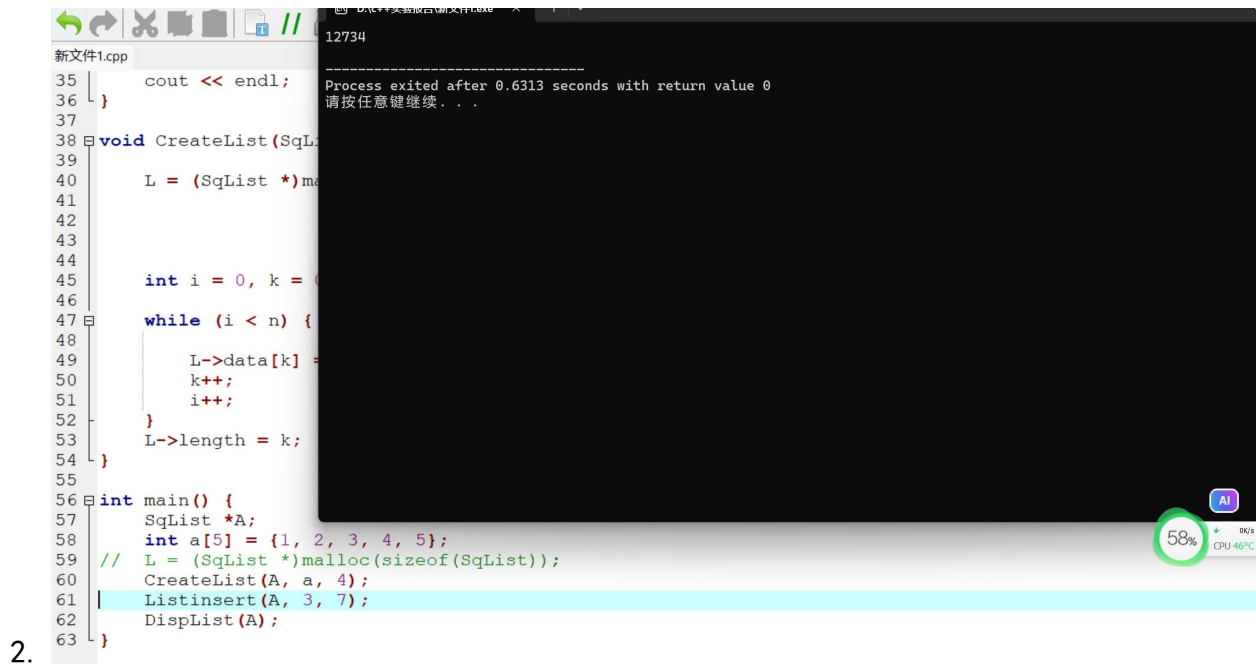
D:\c++实验报告\新文件1.exe

1234

Process exited after 0.6959 seconds with return value 0
请按任意键继续. . .

58% CPU 49°C

2 插入



2.

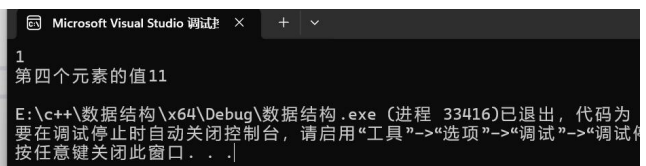
敖元坤：

1. 按序号求线性表元素

```

Initlist(L);
cout << Emptylist(L) << endl;
Creatlist(L, a, 10);
GetElem(L, 4, e);
cout << "第四个元素的值" << e << endl;

```

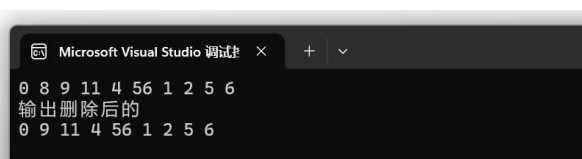


2. 删除

```

//GetElem(L, 4, e);
//cout << "第四个元素的值" << e << endl;
Listdelete(L, 2, e);
cout << "输出删除后的" << endl;

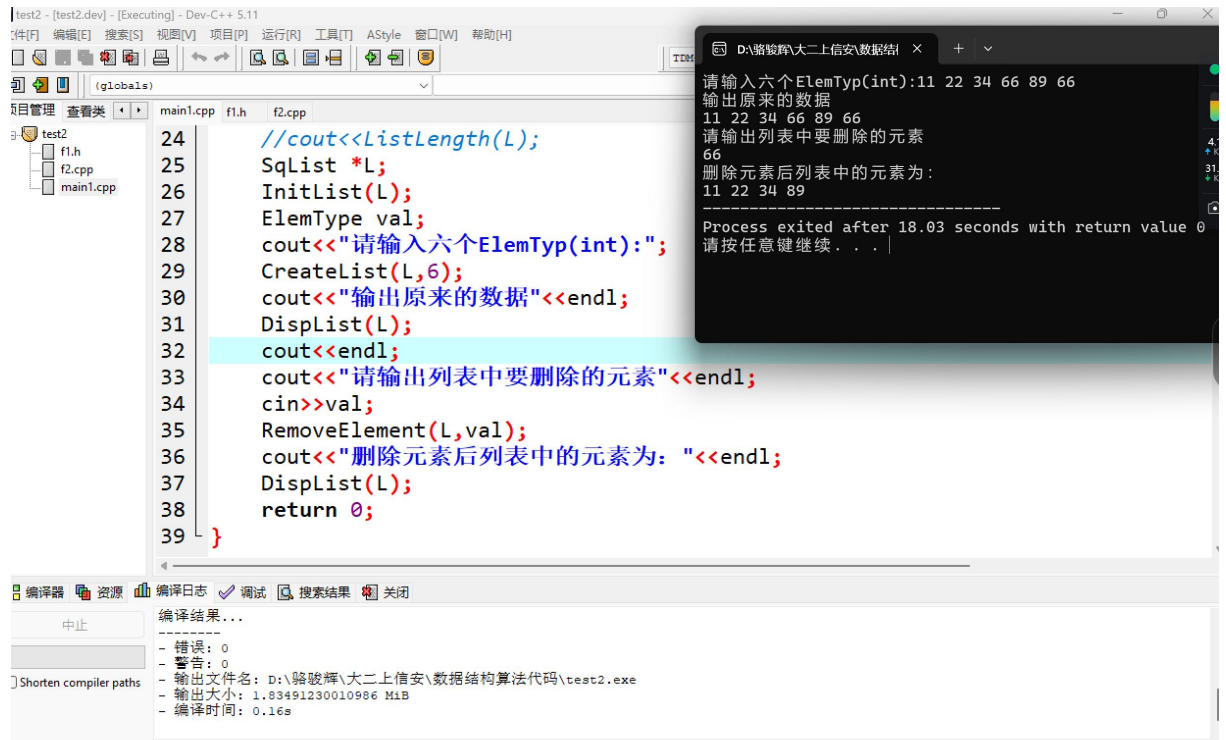
```



三、力扣练习及函数

骆骏辉：

力扣删除多个元素：



The screenshot shows a C++ IDE with a project named 'test2'. The main1.cpp file contains the following code:

```
24 //cout<<ListLength(L);
25 SList *L;
26 InitList(L);
27 ElemType val;
28 cout<<"请输入六个ElemTyp(int):";
29 CreateList(L,6);
30 cout<<"输出原来的数据"<<endl;
31 DispList(L);
32 cout<<endl;
33 cout<<"请输出列表中要删除的元素"<<endl;
34 cin>>val;
35 RemoveElement(L,val);
36 cout<<"删除元素后列表中的元素为: "<<endl;
37 DispList(L);
38 return 0;
39 }
```

The execution output window shows the following text:

```
请输入六个ElemTyp(int):11 22 34 66 89 66
输出原来的数据
11 22 34 66 89 66
请输出列表中要删除的元素
66
删除元素后列表中的元素为:
11 22 34 89
-----
Process exited after 18.03 seconds with return value 0
请按任意键继续...
```

The bottom status bar shows the compilation results:

```
编译结果...
- 错误: 0
- 警告: 0
- 输出文件名: D:\骆骏辉\大二上信安\数据结构算法代码\test2.exe
- 输出大小: 1.83491230010986 Mib
- 编译时间: 0.16s
```

四、集合的运算

黄梓涵：

```

80 int main() {
81     List A, B, C, D;
82     InitList(A);
83     InitList(B);
84
85     // 使用 push_back() 初始化集合A和B的元素
86     A.elements.push_back(1);
87     A.elements.push_back(2);
88     A.elements.push_back(3);
89
90     B.elements.push_back(2);
91     B.elements.push_back(3);
92     B.elements.push_back(4);
93
94     // 进行并集和交集运算
95     unionList(A, B, C);
96     intersectionList(A, B, D);
97
98     // 输出并集 (使用传统的 for 循环)
99     cout << "并集 C: ";
100     for (int i = 0; i < ListLength(C); i++) {
101         cout << C.elements[i] << " ";
102     }
103     cout << endl;
104
105     // 输出交集 (使用传统的 for 循环)
106     cout << "交集 D: ";
107     for (int i = 0; i < ListLength(D); i++) {
108         cout << D.elements[i] << " ";
109     }
110     cout << endl;
111
112     return 0;
113 }

```

C:\Users\34246\Desktop\集合 × + v

```

并集 C: 1 2 3 4
交集 D: 2 3
-----
Process exited after 0.01936 seconds with return value 0
请按任意键继续. . .

```

五、实验小结

(

1. 骆骏辉: 110%
2. 付志城 98%
3. 黄梓涵 98%
4. 王赫权 98%
5. 李祥 98%
6. 敖元坤 98%

完成

骆骏辉: 任务: 完成在顺序表中删除多个与 val 相同的值和求顺序表中的长度。
还完成了额外的力扣题目练习和函数编写

具体问题和解决思路如下

a: 关于如何处理顺序表中删除 val 数据问题:

解决方法: 使用双指针完成, 一个用来读数据, 一个用来写数据, 当读的数据与 val 值不同时写入数据, 相同时则不用写入。

b: 关于编程代码的时候有时会出现漏掉代码结尾的符号问题:

在编写代码时往往有时会少输 “; ” 等符号, 解决方案: 在变成代码的时候要仔细的编写每一句代码尽量做到每一句代码都是有头有尾的。

实验心得: 首先, 在编写一段代码的时候要理清其结构和功能, 编写代码的时候要思考清楚数据所存储的位置, 做到合理的对数据的进行处理, 不然的话会出现数据的错误或实验结果的不准确, 在编写代码的时候也要注意代码的完整性, 及尽量做到代码编写的有始有终, 该加 “; ” 尽量不要少漏。

付志城: 任务: 完成了顺序表的创建和按元素值查找的功能。

具体问题和解决思路如下：

遇到的问题：1 对基础地址理解有所误区 2 创建函数时大小写出错

A:首先一个节点包含的是自己的数据和指向下一个地址的指针，自己对它的理解包含误区，导致自己第一次创建函数的时候，不能够理解它 `s->next=p->next` 的操作，属于是理论概念没弄清楚。自己对项目开始的过程没有很好的理解，导致向队员们发出错误的信息，最后浪费了大量的时间。

B:然后是出现多次创建函数大小写出错的情况，以前命名函数比较随意导致现在数量增多后，我们代码调用的时候经常出错误。

实验心得：要在项目初期开始清晰的任务理解，将任务合理的分配，对实验所需基础知识还要掌握的更加熟练，对项目文件内的关系还需要弄的更加清晰，需要命名的习惯更加良好。

黄梓涵： 任务：完成线性表的交集，补集运算。

在编写代码的过程中，我逐渐意识到自己在编程知识和技能上的不足。这段时间，我不仅在努力解决代码中的问题，还发现了自己对 C++ 语言的理解还不够深入。因此，我开始更多地钻研课本，查阅相关资料，尝试将理论与实践相结合。

课本中关于数据结构、算法和语言特性的内容为我提供了宝贵的指导。我开始认真阅读每一章的细节，反复思考各种概念如何在实际编程中应用。这种深入学习的过程不仅帮助我理解了编程的基本原理，也让我意识到，编程并不是一蹴而就的，而是一个需要不断积累和实践的过程。

通过这种自我反省，我认识到自己在学习上的懈怠，特别是在基础知识的掌握上。每当我在调试代码时遇到挫折，我都会重新翻阅课本，力求理解每一个细节，而不仅仅是找到解决方案。这种转变让我更加重视扎实的基础知识，因为我明白，只有掌握了核心概念，才能在遇到复杂问题时游刃有余。

王赫权：

任务：完成顺序表的逆置和判断顺序表是否为空

具体问题和解决思路如下：

遇到的问题：1. 不清楚实现顺序表逆置的方法； 2. 对数据的交换不知道如何实现。

A. 可以将第一个数据和最后一个数据交换，并依次向后和向前移动标记，来实现所有数据的逆置；

B. 可以定义一个中间变量来存放其中一个数据，防止交换过程中的数据丢失。

实验心得：

在定义一个函数时就要去构思该函数的功能该如何实现，这样才能更顺利地完成整个

函数的编写，否则效率会十分低下。

李祥：任务：完成顺序表的查找和输出功能

遇到的问题：1，不理解 `elemtype` 定义 2，不太理解插入时的操作

a,在写顺序表插入代码时，将新元素插入时把原来元素覆盖了，导致丢失元素。最后理解了应该先将 `i` 后的元素后移一位，再在此位置插入

敖元坤：任务：完成了按序号求线性表中的元素和删除数据元素的功能。

具体问题和解决思路如下：

A：对于函数的调用不够熟练，导致代码在编译过程中出现较多的报错，无法正常运行。

解决方法：通过多渠道的查询，学会函数的正确使用方法，逻辑出现错误就请教小组成员共同解决困难，最终修改好了代码，保证了程序的正常运行。

B：删除数据元素过程中，对于代码的解读错误。

解决方法：去网上搜索相关代码的意义和作用，理解清楚作用之后就能够更加熟练和正确的使用，保证最终程序的正常运行。

实验心得：通过这次实验，认识到了自身的不足之处，同时也感受到了团队的重要性，遇到无法解决的问题要及时的反馈，集思广益共同解决，同时更应该充实自己的技能，在团队中发挥出更好的作用