****

|  |
| --- |
| **面向对象程序设计**  **实验报告**  项目名称  专业班级 联软件2001  学 号 02  姓 名 张圃源  **实验成绩：**  **批阅教师：**  2021 年 11 月 15 日 |
|  |

**实验项目1．类与对象（4学时）**

**1．实验内容**

（1）程序编写：根据问题描述和程序的输出结果，对给出的程序代码进行修改，最终给出自己的解决方案，实际实验中会根据学生课堂实际掌握情况从如下类的程序代码中抽取3~5个：Account，GradeBook，Employee，Date，Complex，SimpleCaculator，IntegerSet等来完成实验。

（2）程序调试：根据给出的存在问题的程序代码，a.修改程序中的编译错误使之能够正确地编译执行；b.对照程序的正确输出结果，修改程序中的逻辑错误使其输出结果和给定的正确输出结果一致。实际实验中会根据学生课堂实际掌握情况从如下类的范例程序中抽取1~3个：Person，Card，Student等来完成实验。

**2．基本要求**

（1）掌握类与对象的定义，掌握类成员的初始化和访问方法；

（2）掌握构造函数和析构函数的声明和不同实现方式；

（3）掌握多文件程序结构的实现方法。

**五、实验说明：**

⑴算法1至算法7可以以头文件的方式存储，主函数实现该头文件的包含即可调用

⑵存储定义

#define MAXSIZE 100 //表中元素的最大个数

typedef int ElemType；//元素类型

typedef struct list{

ElemType elem[MAXSIZE]；//静态线性表

int length； //表的实际长度

}SqList；//顺序表的类型名

⑶建立顺序表时可利用随机函数自动产生数据。

**六、注意问题：**

插入、删除时元素的移动原因、方向及先后顺序。

解不同的函数形参与实参的传递关系。

**七、实验过程及结果**

#include<stdio.h>

#include<stdlib.h>

#define MAXSIZE 20

typedef int ElemType;

typedef struct{

ElemType \*elem;

//int listsize;如果想进行可以扩笼，建议使用此项

int length;

}SqList;

int InitList(SqList \*L)

{

L->elem=(ElemType\*)malloc(sizeof(ElemType)\*MAXSIZE);

if(!L->elem) return 0;

L->length=0;

return 1;

}

int ListInsert(SqList \*L,ElemType e,int i)

{//i表示插入的位置，大小从1到n

if(i<1||i>L->length) return 0;

if(L->length>=MAXSIZE) return 0;

for(int j=L->length;j>=i;j--)

L->elem[j]=L->elem[j-1];

L->elem[i-1]=e;

L->length++;

}

int ListDelete(SqList \*L,ElemType \*e,int i)

{

if(i<1||i>L->length-1) return 0;

if(L->length==0) return 0;

for(int j=i;j<L->length;j++)

L->elem[j-1]=L->elem[j];

\*e=L->elem[i-1];

L->length--;

}

int LocateElem(SqList L,ElemType e){

for(int i=0;i<L.length;i++)

if(L.elem[i]==e) return i+1;

return 0;

}

void InputList(SqList \*L)

{//为了测试简单化，直接赋值

printf("please input the length:");

scanf("%d",&L->length);

printf("\n");

for(int i=0;i<L->length;i++)

L->elem[i]=i+1;

}

void OutputList(SqList L){

printf("the length of sqlist is:%d\n",L.length);

printf("the sqlist is:");

for(int i=0;i<L.length;i++)

printf("%d ",L.elem[i]);

printf("\n");

}

int main()

{

SqList L;

ElemType e;

InitList(&L);

InputList(&L);

OutputList(L);

ListInsert(&L,100,5);

OutputList(L);

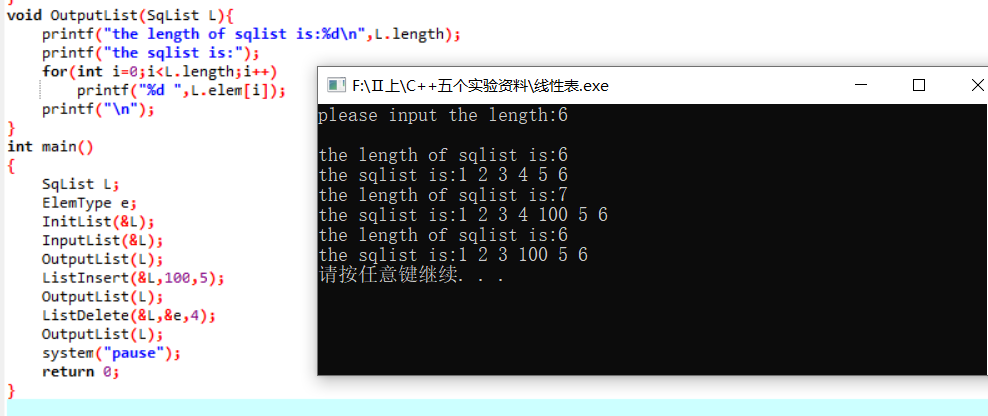
ListDelete(&L,&e,4);

OutputList(L);

system("pause");

return 0;

}



**八、实验小结**

线性表是以一种逻辑结构为线性关系的一组有序数据,即逻辑上元素与元素相邻,至于其元素的存储,这个就是存储结构了,有顺序存储(数组),链式存储。这个实现的线性表的顺序存储结构。

插入元素：判定插入位置是否合适，表是否满，移动元素，插入元素，长度增1。

删除元素：判定删除位置是否合法，移动元素，赋值返回，长度减1。