西安交通大学实验报告

课程名称： 算法设计与问题求解 实验名称： 顺序表基本操作编程

学 院： 机械工程学院 实验日期 2020 年 10 月17 日

班 级： 机械97班 姓 名： 杨逢诜 学号： 2193712613

#### 一、实验内容和结果

* **题目1 实现整型元素线性表**

用数组实现在线性表中插入整数元素，在中间任何位置删除元素，查找一个元素。在main函数中验证。

【源程序】

#include<stdio.h>

const int maxsize=100;

struct SeqList //定义线性表结构及最大表长

{

int data[100];

int length;

};

void Insert(struct SeqList \*L,int i,int x) //声明函数：插入元素

{

int j; //声明计数变量，用于控制循环

if(i>L->length+1||L->length==maxsize||i<0) //若表满或表内对应位置无法存取元素,结束进程并提示

{

printf("ERROR PLACE,CAN'T INSERT IN.\n");

}

else

{

for(j=L->length-1;j>=i-1;j--) //若可以存放元素，则令该位置及其后的全部元素向后平移一个存储单元；

{

L->data[j+1]=L->data[j];

}

L->data[i-1]=x; //空出来的位置用来放指定元素，表长自增

L->length++;

}

}

void Delete(struct SeqList \*L,int i) //声明函数：删除元素

{

int j; //声明计数变量，用于控制循环

if(i>L->length||L->length==0||i<=0) //若表中对应位置没有元素或者表中没有对应位置，结束进程并提示

{

printf("ERROR PLACE,CAN'T DELETE.\n");

}

else

{

for(j=i;j<=L->length;j++) //若该位置有元素，则令该元素后的全部元素向前平移1个存储单元

{

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

}

L->data[L->length-1]=0; //抹掉线性表的最后一个元素，表长自减

L->length--;

}

}

int Find(struct SeqList \*L,int x) //声明函数：线性表查找

{

int j; //声明计数变量，用于控制循环

for(j=0;j<L->length;j++) //从第一个元素开始按顺序找，如果找到某个元素与之相同则返回其下标地址

{

if(L->data[j]==x)

{

return j+1;

}

}

return -1; //若未找到，返回-1

}

void Display(struct SeqList \*L) //声明函数：线性表输出

{

int j; //声明变量，用于控制循环

for(j=0;j<L->length;j++) //从第一个元素开始直至线性表末尾按照格式输出元素

{

printf("The element in place %d is %d.\n",j+1,L->data[j]);

}

}

int main() //主函数开始

{

struct SeqList L; //声明线性表，并置空表

L.length=0;

int s,t,i,id; //声明变量，分别用于存储：插入元素及其位置；控制循环变量；待删除元素；

printf("给出三个有序的待输入整数："); //给出提示，按顺序输入元素，同时将其按顺序插入线性表中；

for(i=0;i<3;i++)

{

scanf("%d",&s);

Insert(&L,i+1,s);

}

Display(&L); //输出表，用于检验

printf("给出待删除的整数："); //给出提示，输入待删除元素，并寻找其位置

scanf("%d",&id);

if(Find(&L,id)!=-1) //若能找到该元素，则将其删除

{

Delete(&L,Find(&L,id));

}

Display(&L); //输出表，用于检验

printf("给出一个待插入的整数和插入的位置："); //给出提示，在指定位置再输入一个元素，并插入线性表中；

scanf("%d %d",&s,&t);

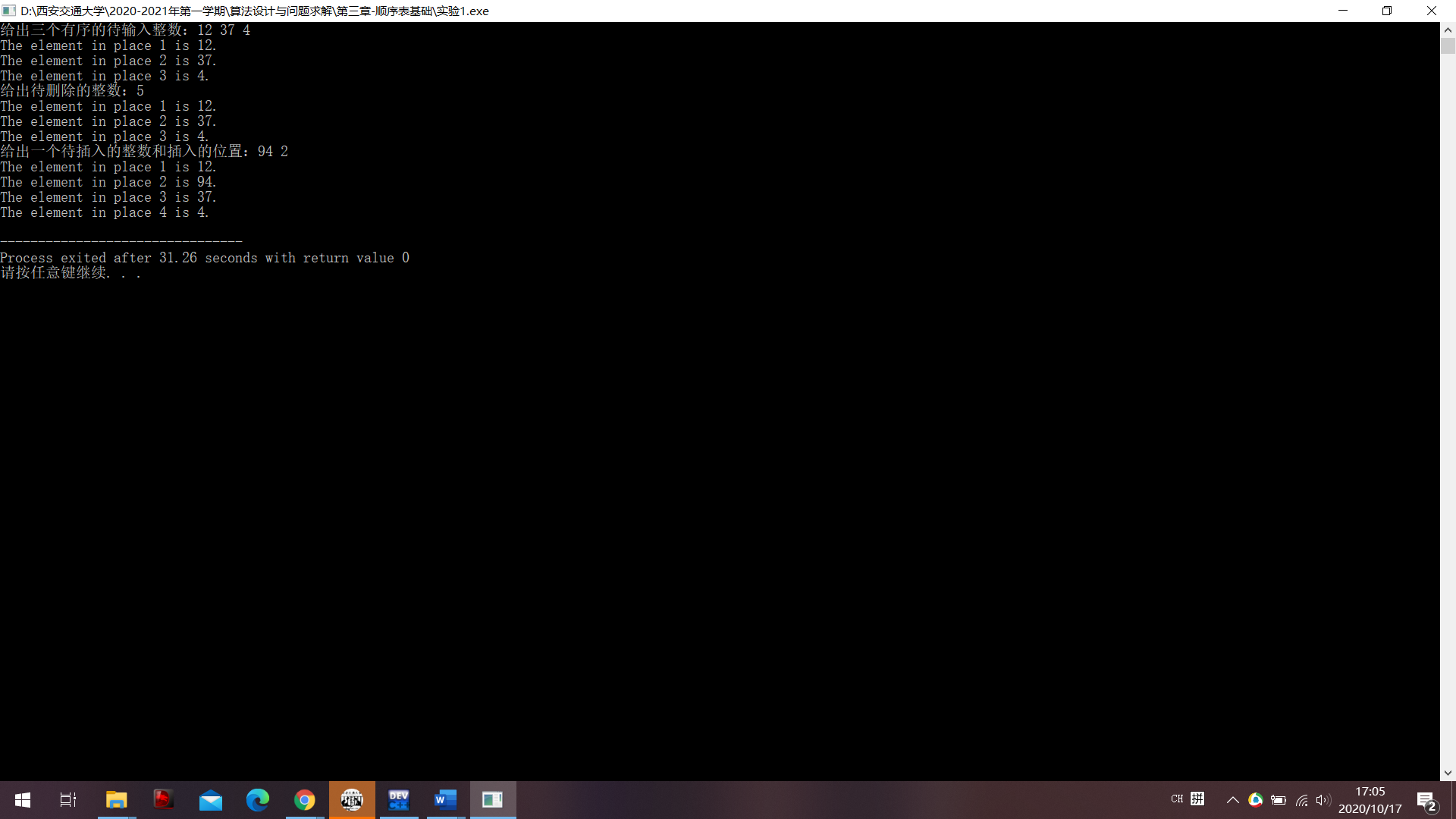
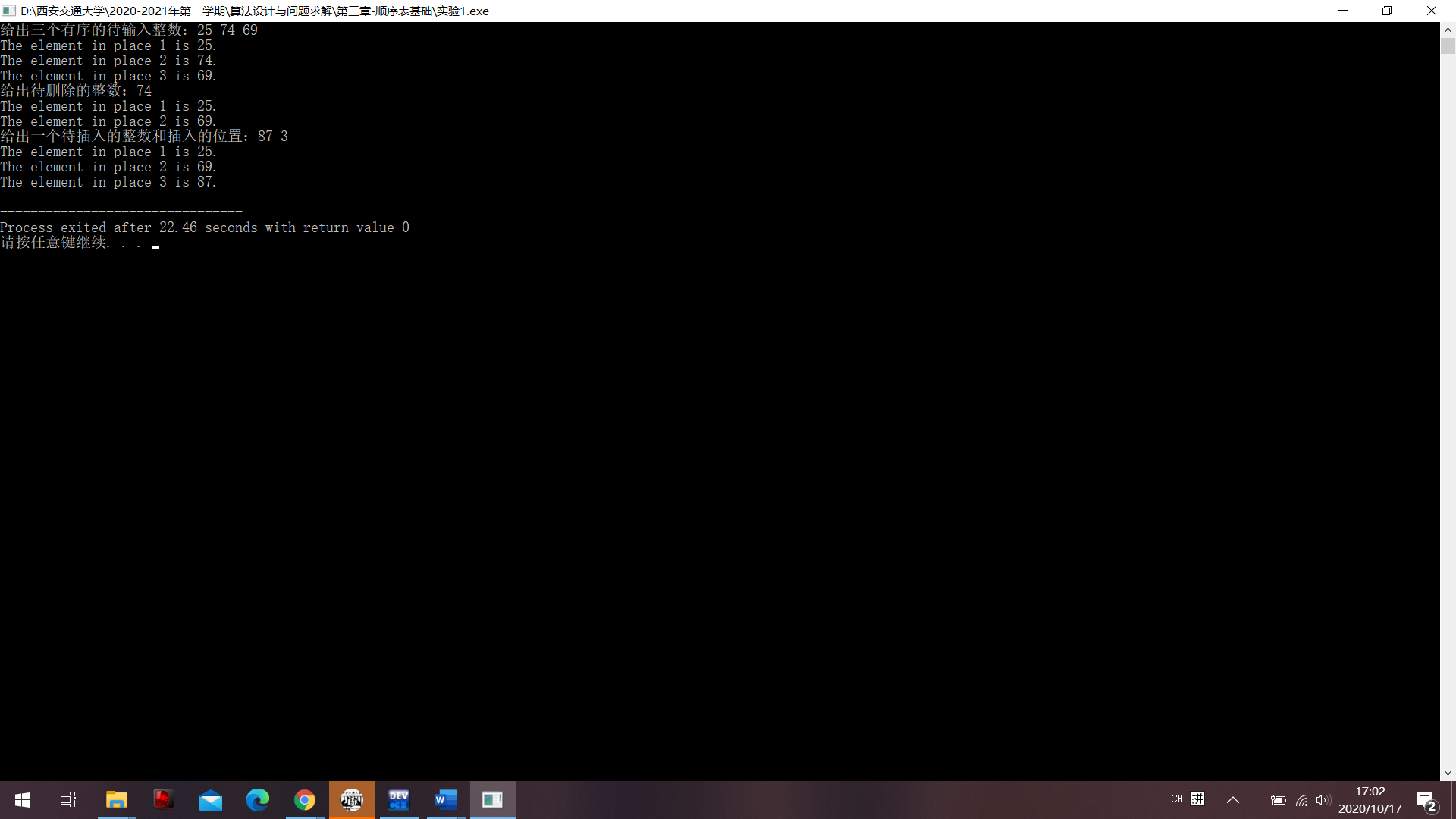
Insert(&L,t,s);

Display(&L); //输出表，主函数结束

return 0;

}

【运行结果】



* **题目2 学生信息管理线性表**

将实验1中的元素用结构体代替，实现程序。

要求：

（1）插入学生记录时，学号按从小到大顺序插入

（2）删除学生信息时，按学号删除

【源程序】

#include<stdio.h>

const int maxsize=100; //构建结构体变量：学生数据单元；

struct stu

{

int id;

char name[20];

};

struct SeqList //构建线性表并定义其最大表长

{

struct stu data[100];

int length;

};

void InsertInOrder(struct SeqList \*L,struct stu info) //声明函数：元素插入

{

int j; //声明计数变量，用于控制循环

if(L->length==maxsize) //若表满，则终止进程并提示

{

printf("NO PLACE TO STORE.");

}

else if(L->length==0) //若表空，则直接填上第一个元素并使表长自增

{

L->data[0]=info;

L->length++;

}

else

{

for(j=L->length;;j--) //其他情形下，则执行循环进程：

{

if(j==0) //若循环进行到了线性表表头，则可知该学生学号为最小，将元素直接填入表头

{

L->data[j]=info;

break;

}

if(info.id>L->data[j-1].id) //若找到了一个学号恰好小于该元素的元素，则将待插入元素放在这个元素后面，同时终止循环；

{

L->data[j]=info;

break;

}

else //其他情形下，让学号大的学生信息向后平移一个存储单元

{

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

}

}

L->length++; //表长自增

}

}

void DeleteInID(struct SeqList \*L,int id) //声明函数：元素删除

{

int j,i=0; //声明计数变量，用于控制循环

for(j=0;j<L->length;j++) //从第一个元素起，到最后一个元素为止，查找学生；

{

if(id==L->data[j].id) //如果找到了对应学号的学生，则将其后全部学生的信息全部向前平移一个存储单元，并使变量I自增以指示“确实删掉了一个”

{

int k=j;

for(;k<L->length-1;k++)

{

L->data[k]=L->data[k+1];

i++;

}

}

}

printf("%s",i==0?"NO SUCH STUDENT.\n":"INFO IS DELETED.\n");

if(i!=0) //如果我们确实删掉了一个元素，则令表长自减并提示；否则则提示找不到对应学生

{

L->length--;

}

}

void Display(struct SeqList \*L) //声明函数：线性表输出

{

int j; //声明计数变量，用于控制循环

for(j=0;j<L->length;j++) //从第一个元素起，到最后一个元素为止，按照格式输出全部元素

{

printf("The element in place %d is %d %s.\n",j+1,L->data[j].id,L->data[j].name);

}

}

int main() //主函数开始

{

int i; //声明计数变量，用于控制循环

struct SeqList L; //声明线性表并置空表

L.length=0;

struct stu stuinfo; //声明结构体用于暂时存放待输入元素

printf("给出三个待输入信息："); //按照提示输入学生信息并将其依次放入线性表中

for(i=0;i<3;i++)

{

scanf("%d ",&stuinfo.id);

gets(stuinfo.name);

InsertInOrder(&L,stuinfo);

}

Display(&L); //输出表，用于检测

printf("给出待删除的信息："); //按照提示输入学生学号并将其依次放入线性表中

scanf("%d",&stuinfo.id);

DeleteInID(&L,stuinfo.id); //删掉学号对应的学生信息，并输出表

Display(&L);

printf("给出一个待插入信息："); //再插入一个学生信息，按照提示输入学生信息并将其放入线性表中，最终输出表，主函数结束

scanf("%d ",&stuinfo.id);

gets(stuinfo.name);

InsertInOrder(&L,stuinfo);

Display(&L);

return 0;

}

【运行结果】

