一·实验目的。

成绩：

指导教师：

（签字）：

计算机工程系实验报告

课程名称：线性表

实验项目名称： 顺序表的插入与删除

班级： 姓名： 学号：

|  |  |  |
| --- | --- | --- |
| 网络三班 | 张乐乐 | 160902336 |

1. 学习顺序表基本操作与基本的知识；
2. 了解顺序表存储方式与插入与删除操作；
3. 了解表格变化是存储的位置变化；

二·实验要求。

1. 将主函数与各子函数分离分别进行模块化编写；
2. 以结构体进行定义函数并操作；

三·完成程序。

#include <stdio.h>

#include <stdlib.h>

#define LIST\_INIT\_SIZE 5 //线性表初始长度

#define LISTINCREMENT 5 //线性表每次扩充长度

#define OK 1

#define ERROR 0 //线性表结构体

typedef struct SqList{

int \*elem;

int len;

int size;

};

//初始化线性表

int InitList\_Sq(struct SqList \*L){

//使用malloc函数，申请一段连续的存储空间

L->elem = (int \*)yi(LIST\_INIT\_SIZE \* sizeof(int));

if(!L->elem)exit(0);

L->len=0;

L->size = LIST\_INIT\_SIZE;

return OK;

}

//向线性表中插入数据

int ListInsert\_Sq(struct SqList \*L,int i,int e){

if(i<1||i>L->len+1) return ERROR;

if(L->len>=L->size){

//使用realloc函数，在原本的基地址上申请更大的存储空间

int \*newbase = (int \*)realloc(L->elem,(LIST\_INIT\_SIZE + LISTINCREMENT) \* sizeof(int));

//如果申请空间失败则结束程序

if(!newbase)exit(0);

L->size += LISTINCREMENT;

}

int \*q = &(L->elem[i-1]);

int \*p;

for(p=&L->elem[L->len-1];p>=q;--p)

\*(p+1)=\*p;

\*q=e;

++L->len;

return OK;

}

//删除线性表中的数据

int ListDelete\_Sq(struct SqList \*L,int i,int \*e){

if(i<1||i>L->len) return ERROR;

int \*p=&(L->elem[i-1]);

\*e=\*p;

int q= L->elem + L->len-1;

for(++p;p<=q;++p)

\*(p-1)=\*p;

L->len--;

return OK;

}

//输出线性表中的数据

int ListShow\_Sq(struct SqList \*L,char\* s){

printf("%s",s);

int i;

for(i=0;i<L->len;i++){

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

}

putchar('\n');

}

void main(){

int i;

struct SqList L;

//初始化线性表

InitList\_Sq(&L);

//输入数据

int n;

printf("预先输入的原始数据个数是：");

scanf("%d",&n);

printf("请线性表中%d个原始数据：",n);

for(i=0;i<n;i++){

int t;

scanf("%d",&t);

ListInsert\_Sq(&L,L.len+1,t);

}

ListShow\_Sq(&L,"原始数据：");

//定义插入/删除操作时位置和数值参数

int s,v;

//插入操作

printf("请输入数据插入的位置s 和数值v ：");

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

printf("%s",ListInsert\_Sq(&L,s,v)?"插入成功.\n":"插入失败.\n");

ListShow\_Sq(&L,"插入后：");

//删除操作

printf("请输入数据删除的位置s ：");

scanf("%d",&s);

if(ListDelete\_Sq(&L,s,&v))

printf("删除成功.删除的数据是：%d\n",v);

else

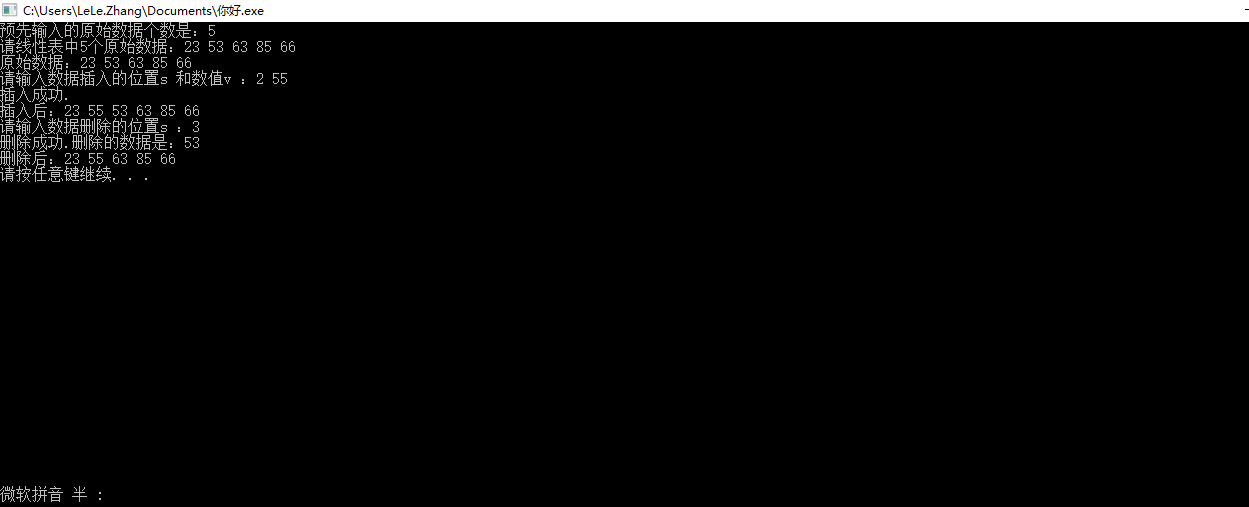
printf("删除失败.位置有误.");

ListShow\_Sq(&L,"删除后：");

system("pause");

}

四·实验抓图与分析



五·实验错误分析。

1. 创建线性表时注意以malloc申请空间；
2. 插入时判断线性表是否空或满；
3. 删除时判断线性表是否空；

六·附录。

参考文献： 耿国华《数据结构—用C语言描述》