实验 数据结构(顺序表的建立于操作)

**[实验目的]**理解数据结构的顺序存储关系，掌握基本操作算法；

**[实验环境**] VC6.0 或VS2010/2012

**[实验步骤]**按要求建立C源程序，观察并分析结果。

说明：以下程序构造顺序表SeqList ，并且定义了顺序表的常用操作，主函数依次调用这些函数，完成相应操作。

**源程序如下：**

#include "stdio.h"

#include "stdlib.h"

/\*\*\*\*以下定义常量,\*\*\*\*/

#define OK 1(这几步代表给OK赋值1并且OK的值永远不变。)

#define ERROR 0

#define TRUE 1

#define FALSE 0

#define MAXSIZE 20 //顺序表最大长度

/\*以下定义类型的同义字\*/

typedef int Status; //凡遇到Status即为int类型

typedef int ElemType; //凡遇到ElemType即为int类型

/\*以下定义顺序表SeqList\*/

typedef struct(创建一个表格)

{

ElemType data[MAXSIZE]; //data用于存放表结点

int length; // 表长度

}SeqList;

# /\*\*顺序表初始化\*\*/

Status InitSeqList(SeqList \*seqList) // \*seqListL表示指向顺序表SeqList的指针

{

seqList->length=0; //置为空表

return OK;

}

# /\*求顺序表的长度\*/

int GetSeqListLen(SeqList \*seqList)

{

return seqList->length;

}

# /\*插入元素(在第n个元素之前的位置插入新元素)\*/

Status Insert(SeqList \*seqList,int n,ElemType e)

{

int k;

//检查数组是否已满

if (seqList->length>=MAXSIZE) return ERROR;

//检查n的位置是否超出范围

if (n<1||n>seqList->length+1) return ERROR;

//若插入数据位置不在表尾

if (n<=seqList->length)

{

//将要插入位置之后元素依次向后移动一位

for (k=seqList->length-1;k>=n-1;k--)

{

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

}

}

//将新元素插入到腾出的位置，并将表长加1

seqList->data[n-1]=e;

seqList->length++;

return OK;

}

# /\*删除元素(删除第n个元素)\*/

Status Delete(SeqList \*seqList,int n,ElemType \*e)

{

int k;

//判断数组是否为空

if (seqList->length==0) return ERROR;

//判断n的位置是否合法

if (n<1||n>seqList->length) return ERROR;

\*e=seqList->data[n-1];

//如果删除不是最后位置

if (n<seqList->length)

{

//将删除位置后继元素依次前移

for (k=n;k<seqList->length;k++)

{

seqList->data[k-1]=seqList->data[k];

}

}

//表长减1

seqList->length--;

return OK;

}

# /\*查找第n个元素\*/

int GetDataByIndex(SeqList \*seqList,int n)

{

//检查位置是否超出范围

if (n<1||n>seqList->length) return ERROR;

return seqList->data[n-1];

}

/\*打印结果\*/

void Display(SeqList \*seqList)

{

int i;

printf("\n\*\*\*\*\*\*\*\*展示数据\*\*\*\*\*\*\*\*\n");

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

{

printf("%d\n",seqList->data[i]);

}

printf("\n");

}

/\*\*\*\*\*\*\*\*\*以下是主程序\*\*\*\*\*\*\*\*/

void main()

{

//声明变量

SeqList seqList; //创建顺序表

int i,j,k,m,elem; //声明变量

printf("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*初始化\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

i=InitSeqList(&seqList);// 调用InitSeqList函数，其中&符号表示地址

printf("初始化后表的长度为：%d\n",seqList.length);

printf("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*插入五条数据\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

for (j=1;j<=5;j++)

{

printf("输入第%d条数据：",j);

scanf("%d",&m);

i=Insert(&seqList,j,m);//调用Insert函数，在表头依次插入5个数

}

Display(&seqList); //调用Display函数，显示顺序表

printf("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*删除一条数据\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("删除哪条数据？");

scanf("%d",&j);

i=Delete(&seqList,j,&elem); //调用Delete函数，删除指定数据

if (i==OK) printf("删除成功\n");

Display(&seqList); //调用Display函数，显示顺序表

printf("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*按位置查找元素\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("查找第几条数据？");

scanf("%d",&j);

k=GetDataByIndex(&seqList,j); //调用GetDataByIndex函数

printf("\n第%d个元素为%d\n",j,k);

printf("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*获取表的当前长度\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

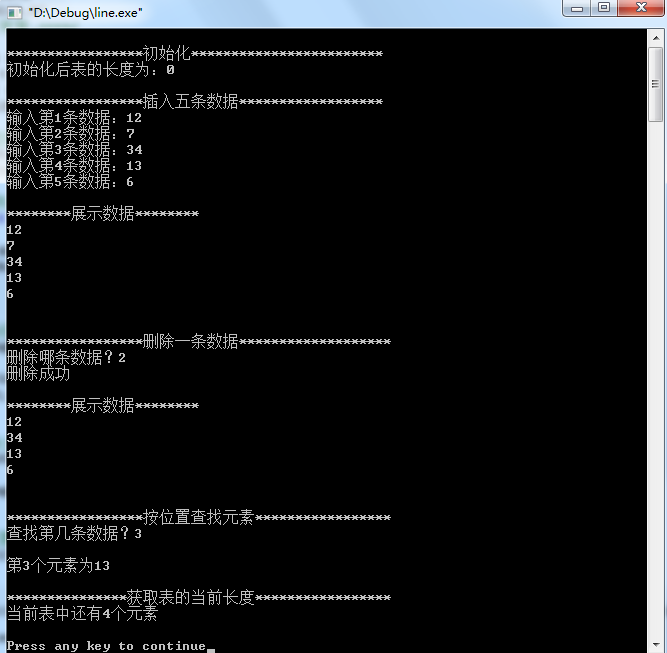
k=GetSeqListLen(&seqList); //调用GetSeqListLen函数

printf("当前表中还有%d个元素\n\n",k);

getchar();

}

程序运行结果：



# 实验提交要求:

1. 写出顺序表的插入、删除、查找第n个记录的过程。

插入：（1）检查数组是否已满

（2）检查n的位置是否超出范围

（3）若插入数据位置不在表尾，将要插入位置之后元素依次向后移动一位或将新元素插入到腾出的位置，并将表长加1.

删除： （1）判断数组是否为空.

（2）判断n的位置是否合法

（3） 如果删除不是最后位置，将删除位置后继元素依次前移并将表长减1。

查找：（1）检查位置是否超出范围.

（2）打印结果

（3）声明变量

（4）创建数据表

（5）声明变量

（6）调用InitSeqList函数，其中&符号表示地址

（7）调用Insert函数，在表头依次插入5个数

（8）调用Display函数，显示顺序表

（9）调用Delete函数，删除指定数据

（10）调用Display函数，显示顺序表

（11）调用GetDataByIndex函数

（12）调用GetSeqListLen函数

