通信与信息工程学院

《数据结构》实验报告

|  |  |
| --- | --- |
| 班 级： | 电信1702 |
| 姓 名： | 秦伟钦 |
| 学 号： | 17407111714 |
| 指导教师： | 刘晓佩 |
| 实验时间： |  |

通信与信息工程学院

二〇一九年

**目录**

实验1 线性表的基本操作---------------------------------------------------------------------------------------------页码1

**完成所有实验报告后，按照上述格式形成目录。**

## 实验一 线性表的基本操作

## 一、实验目的

1. 熟悉C/C++语言上机环境；

2. 掌握线性表的基本操作：查找、插入、删除等运算在顺序存储、链式存储结构上的运算。

## 二、实验环境

1. 装有Visual C＋＋6.0的计算机。

2. 本次实验共计2学时。

## 三、实验内容

1. 建立顺序表，基本操作包括：**建立并初始化顺序表、输出顺序表、判断是否为空、取表中第i个元素、查找、插入和删除**。并在主函数中完成对各种函数的测试。

测试用例为：L=（3,6,9 ,0, -2,89,77）

1. 设有两个非递增有序的线性表A和B，均已顺序表作为存储结构。编写算法实现将A表和B表合并成一个非递增有序排列的线性表（可将线性表B插入线性表A中，或重新创建线性表C）。

测试用例： A = ( 29, 18, 12, 7, 6)

B= ( 33, 22, 17,12,5)

合并后的线性表C=(33, 29, 22, 18, 17, 12, 12, 7, 6, 5)

3. 建立单链表，基本操作包括：初始化、判断是否为空、取表中第i个元素、查找、插入和删除。并在主函数中完成对各种函数的测试。

## 四、源程序

#include<string.h>

#include<ctype.h>

#include<malloc.h>

#include<limits.h>

#include<stdio.h>

#include<stdlib.h>

#include<io.h>

#include<math.h>

#include<process.h>

/\* 函数结果状态代码 \*/

#define TRUE 1

#define FALSE 0

#define OK 1

#define ERROR 0

#define INFEASIBLE -1

/\* #define OVERFLOW -2 因为在math.h中已定义OVERFLOW的值为3,故去掉此行 \*/

typedef int Status; /\* Status是函数的类型,其值是函数结果状态代码，如OK等 \*/

typedef int Boolean; /\* Boolean是布尔类型,其值是TRUE或FALSE \*/

#define LIST\_INIT\_SIZE 100

#define LISTINCREMENT 10

typedef int ElemType ;

typedef struct{

ElemType \*elem;

int length;

int listsize;

}SqList;

Status Initlist\_Sq(SqList &L){

L.elem=(ElemType \*)malloc(LIST\_INIT\_SIZE\*sizeof(ElemType));

if(!L.elem)exit(OVERFLOW);

L.length=0;

L.listsize=LIST\_INIT\_SIZE;

return OK;

}

//构造一个空的线性表

Status ListInsert\_Sq(SqList &L,int i,ElemType e){

if(i<1||i>L.length+1) return ERROR;

if(L.length>=L.listsize){

ElemType \*newbase=(ElemType \*)realloc(L.elem,(L.listsize+LISTINCREMENT)\*sizeof(ElemType));

if(!newbase)exit(OVERFLOW);

L.elem=newbase;

L.listsize+=LISTINCREMENT;

}

ElemType \*p,\*q=L.elem+i-1;

q=&(L.elem[i-1]);

for(p=&(L.elem[L.length-1]);p>=q;--p)\*(p+1)=\*p;

\*q=e;

++L.length;

return OK;

}

//线性表插入

Status ListDelete\_Sq(SqList &L,int i,ElemType &e){

if(i<1||i>L.length) return ERROR;

ElemType \*p,\*q;

p=&(L.elem[i-1]);

e=\*p;

q=L.elem+L.length-1;

for(++p;p<=q;++p)\*(p-1)=\*p;

--L.length;

return OK;

}

//删除

Status GetElem(SqList L,int i,ElemType &e){

if(i<1||i>L.length)

return ERROR;

ElemType \*p;

p=&(L.elem[i-1]);

e=\*p;

return OK;

}

//查找线性表里的第i个元素

Status DestroyList(SqList &L){

free(L.elem);

L.elem=NULL;

L.length=0;

L.listsize=0;

return OK;

}

//线性表的销毁

Status ClearList(SqList &L){

L.length=0;

return OK;

}

//清空线性表

bool ListEmpty(SqList L){

return (L.length==0)?true:false;

}

//判空线性表

bool ListFull(SqList L){

return (L.length==L.listsize)?true:false;

}

//判满线性表

void print(SqList L){

for(int i=1;i<L.length+1;i++)

printf("%d ",L.elem[i-1]);

}//输出

int ListLength(SqList L){

return L.length;

}

Status compare(ElemType e1,ElemType e2){

if(e1==e2)

return 1;

if(e1<e2)

return -1;

return 0;

}

int LocateElem(SqList L,ElemType e,Status(\*compare)(ElemType,ElemType)){

int i=1;

while(i<L.length&&compare(L.elem[i-1],e)){

i++;

if(i==L.length+1)

return ERROR;

else return i;

}

}

void Combine(SqList &A,SqList &B){

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

int count=0;

for(int j=0;j<A.length;j++){

if(A.elem[j]==B.elem[i]){

count+=1;

}

}

if(count==0){

A.elem[A.length++]=B.elem[i];

}

}

}

void mainview\_user( ) //界面函数

{ SqList La;

SqList Lb;

Initlist\_Sq(La);

Initlist\_Sq(Lb);

int c;

while(1)

{

system("CLS"); //清除屏幕函数

printf(" ------------------------------------\n");

printf(" |\*\*\*\*\*\*\*\*\*\*线性表\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*|\n");

printf(" |\*\*\*\*\*\*\*\*1 输入数据\*\*\*\*\*\*\*\*\*\*\*|\n");

printf(" |\*\*\*\*\*\*\*\*2 查看数据\*\*\*\*\*\*\*\*\*\*\*|\n");

printf(" |\*\*\*\*\*\*\*\*3 删除数据\*\*\*\*\*\*\*\*\*\*\*|\n");

printf(" |\*\*\*\*\*\*\*\*4 查找数据\*\*\*\*\*\*\*\*\*\*\*|\n");

printf(" |\*\*\*\*\*\*\*\*5 销毁线性表\*\*\*\*\*\*\*\*\*|\n");

printf(" |\*\*\*\*\*\*\*\*6 清空线性表\*\*\*\*\*\*\*\*\*|\n");

printf(" |\*\*\*\*\*\*\*\*7 改数据 \*\*\*\*\*\*\*\*\*|\n");

printf(" |\*\*\*\*\*\*\*\*8 插入数据\*\*\*\*\*\*\*\*\*\*\*|\n");

printf(" |\*\*\*\*\*\*\*\*9 线性表合并\*\*\*\*\*\*\*\*\*|\n");

printf(" |\*\*\*\*\*\*\*\*0 退出系统\*\*\*\*\*\*\*\*\*\*\*|\n");

printf(" ------------------------------------\n");

printf("\n");

printf("请选择：");

scanf("%d",&c);

switch(c)

{

case 1: {

int b,a;

printf("要输入多少个数据:");

scanf("%d",&b);

for(int i=0;i<b;i++){

scanf("%d",&a);

ListInsert\_Sq(La,i+1,a);

}

printf("数据为：");

print(La);

break;

}

case 2: {

print(La);

break;

}

case 3: {

int b,a3;

print(La);

printf("\n");

printf("请输入删除第几个数据:");

scanf("%d",&b);

if(b<1||b>La.length)

printf("数据非法");

else{

ListDelete\_Sq(La,b,a3);

printf("删除的数据为：");

printf("%d\n",a3);

printf("删除成功");

print(La);

}

break;

}

case 4: {

int b,a3,c[20],j=0;

memset(c,0,sizeof(c));

printf("请输入查找的元素：");

scanf("%d",&b);

for(int i=1;i<La.length+1;i++){

{if( La.elem[i-1]==b)

c[j++]=i;

}

}

if(c[0]==0)

printf("没有此元素！");

else{

printf("%d的位置为：",b);

for(int k=0;c[k]!=0;k++)

printf("%d ",c[k]);}

break;

}

case 5: {

DestroyList(La);

break;

}

case 6:{

ClearList(La);

break;

}

case 7:{

print(La);

printf("\n");

printf("请输入想要修改的位置:");

int a,a3,b;

scanf("%d",&a);

if(a<1||a>La.length)

printf("数据非法");

else{

ListDelete\_Sq(La,a,a3);

printf("请输入修改的数据:");

scanf("%d",&b);

ListInsert\_Sq(La,a,b);

printf("修改后为：");

print(La);

printf("\n");

printf("修改成功");}

break;

}

case 8:{

int i,b;

print(La);

printf("\n");

printf("输入插入的位置:");

scanf("%d",&i);

if(i<1||i>La.length)

printf("数据非法");

else {

printf("插入的元素:");

scanf("%d",&b);

ListInsert\_Sq(La,i,b);

printf("插入成功");

print(La);

}

break;

}

case 9:{

int a,b;

printf("线性表A有多少数据：");

scanf("%d",&a);

for(int i=0;i<a;i++){

scanf("%d",&b);

ListInsert\_Sq(La,i+1,b);

}

printf("线性表B有多少数据：");

scanf("%d",&a);

for(int i=0;i<a;i++){

scanf("%d",&b);

ListInsert\_Sq(Lb,i+1,b);

}

Combine(La,Lb);

printf("合并为：");

print(La);

break;

}

case 0: {

DestroyList(La);

return;

}

default:printf("输入错误，请重新输入!\n");fflush(stdin);

}

printf("\n\n");

system("PAUSE");

}

}

int main()

{

mainview\_user( );

}

总结和体会：

刚开始学的时候确实有很多地方我很不理解，每次上课时老师都会给我们出不同的设计题目，对于我们一个初学者来说，无疑是一个具大的挑战，撞了几次壁之后，我决定静下心来，仔细去写程序。老师会给我们需要编程的内容一些讲解，顺着老师的思路，来完成自己的设计，我们可以开始运行自己的程序，可是好多处的错误让人看的可怕，还看不出到底是哪里出现了错误，但是程序还是得继续下去，我多次请教了老师和同学，逐渐能自己找出错误，并加以改正。TC里检查错误都是用英文来显示出来的，经过了这次课程设计，现在已经可以了解很多错误在英文里的提示，这对我来说是一个突破性的进步，眼看着一个个错误通过自己的努力在我眼前消失，觉得很是开心。