数据结构实验报告

院系：信息工程学院

专业：软件工程二班

学号：20122480212

姓名：霍静静

**实验一：顺序表的基本操作**

**一、实验目的**：了解顺序表的逻辑结构特性，熟练掌握顺序表存储结构的描述方法，熟练掌握顺序表的基本操作：查找、插入等

**二、实验要求**：

1）建预定义常量和类型的头文件

2）定义顺序表类型

3）创建顺序表函数，实现几个基本操作

4）写函数完成实验内容要求。

5）写出主函数调用，调试运行程序完成整个实验，显示结果。

**三、实验内容：**

1、 实现2个有序顺序表的归并

**四、算法思想：**

线性表的顺序表示指的是用一组地址连续的存储单元一次存储线性表的数据元素，基本操作就是对这些数据进行操作。

**五、主要函数的实现：**

#include<stdio.h>

#include<stdlib.h>

#define OK 1

#define ERROR 0

#define TURE 1

#define FALSE 0

#define OVERFLOW -2

#define LIST\_INIT\_SIZE 100

#define LISTINCREMENT 10

typedef int elemType;

typedef int Status;

typedef struct

{

elemType \*elem;

int length;

int listsize;

}sqList;

Status initList\_Sq(sqList &L);//构造一个空的线性表

Status inputList(sqList &L,int n);//在线性表L中添加n个元素

Status listInsert\_Sq(sqList &L,int i,elemType e);//在顺序表L中第i个位值之前插入新的元素e

Status listDelete\_Sq(sqList &L,int i,elemType &e);//在顺序线性表中删除第i个元素，并用e返回

void mergeList\_Sq(sqList La,sqList Lb,sqList &Lc);//一直顺序线性表La和Lb的元素按非递减排列，

//归并La和Lb得到新的顺序线性表Lc，Lc的元素也按非递减顺序排列

void outputList(sqList L);//输出顺序线性表中的所有元素

//--------主函数----------------------------

int main()

{

sqList la,lb,lc;

int n;

initList\_Sq(la);initList\_Sq(lb);

printf("请输入链La的元素个数：");

scanf("%d",&n);

inputList(la,n);

printf("请输入链Lb的元素个数：");

scanf("%d",&n);

inputList(lb,n);

mergeList\_Sq(la,lb,lc);

outputList(lc);

getchar();getchar();getchar();getchar();

return 0;

}

//-------子函数-----------------------------

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;

}

Status inputList(sqList &L,int n)

{

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

{

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

L.length++;

}

return 0;

}

Status listInsert\_Sq(sqList &L,int i,elemType e)

{

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

if(L.length>=L.listsize)

{

elemType \*newBase = (elemType \*)realloc(L.elem,(L.listsize+LISTINCREMENT)\*sizeof(elemType));

if(! newBase) return ERROR;

L.elem = newBase;

L.listsize +=LISTINCREMENT;

}

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

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

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

\*q = e;

++L.length;

return OK;

}

Status listDelete\_Sq(sqList &L,int i,elemType &e)

{

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

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

e = \*p;

elemType \*q =L.elem+L.length-1;

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

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

--L.length;

return OK;

}

void mergeList\_Sq(sqList La,sqList Lb,sqList &Lc)

{

elemType \*pa = La.elem;

elemType \*pb = Lb.elem;

Lc.listsize = Lc.length = La.length+Lb.length;

elemType \*pc = Lc.elem = (elemType \*)malloc(Lc.listsize\*sizeof(elemType));

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

elemType \*pa\_last = La.elem + La.length - 1;

elemType \*pb\_last = Lb.elem + Lb.length - 1;

while(pa<=pa\_last&&pb<=pb\_last)

{

if(\*pa <= \*pb) \*pc++ = \*pa++;

else \*pc++ = \*pb++;

}

while(pa<=pa\_last) \*pc++ = \*pa++;

while(pb<=pb\_last) \*pc++ = \*pb++;

}

void outputList(sqList L)

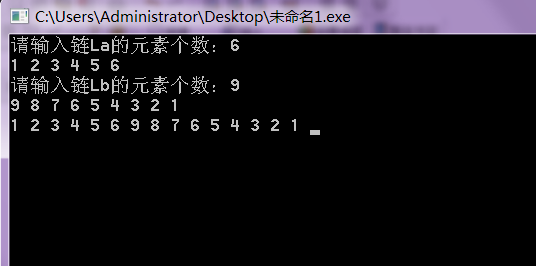
{

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

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

}

**六、测试结果：**

****

**七、实验总结：**

**1.还不会用头文件，不能构成一个线性表的头文件；**

**2.对指针的运算还有很大的欠缺；**