安 徽 新 华 学 院 实 验 报 告

姓 名 张卢 学号 1632104249 专业班级 16软件2班

课程名称 数据结构 实验名称 顺序表的基本操作的实现 实验日期 2017.9.25 同组人员 指导老师 汪红霞 得分

**实 验 项 目 二**

**【实验名称】**顺序表的基本操作的实现

**【实验类型】**基础性实验

**【实验目的与要求】**

* 掌握线性表的基本操作在顺序存储上的实现；
* 掌握线性表顺序存储的各种操作（建立、插入、删除等）的实现；
* 熟练C/C++语言的使用（特别是函数参数、数组的使用）。

**【实验主要仪器设备及台套数】**

一人一台安装Visual C++的电脑

**【实验学时】** 4学时

**【实验内容】**

1. 实现顺序表存储的各种操作（建立、插入、删除、查找、显示、求长度、求前驱、求后继等）的实现。

#include <iostream.h>

const int maxsize=10;

typedef int elemtype;

class sequenlist

{

public:

elemtype a[maxsize]; //表示线性表(a1,a2,…,an,aMaxlen), elemtype表示某种具体数据类型

int len; //len表示线性表的实际长度

public:

int length(sequenlist L ); //求顺序表L的长度

void Insert(sequenlist &L,elemtype x,int i);//将元素x插入顺序表L第i个位置

void Dele(sequenlist &L , int i) ; //删除顺序表L第i个位置的元素

void setnull(sequenlist &L ); //顺序表L置空表

int Locate(sequenlist L , elemtype x) ; //定位，顺序表L中查找元素x的位置

elemtype Get(sequenlist L , int i); //取顺序表L中第i个位置的元素

elemtype prior(sequenlist L , elemtype x ); //顺序表L中求元素x的前驱

elemtype next(sequenlist L , elemtype x ); //顺序表L中求元素x的后继

void prin(sequenlist L);//输出线性表中的元素

};

elemtype sequenlist ::Get(sequenlist L,int i)

{

if((i<1)||(i>L.len)) return NULL;

else return L.a[i];

}

elemtype sequenlist :: prior(sequenlist L,elemtype x)

{

int j=1;

while((j<=L.len)&&(L.a[j]!=x))

j++;

if(j<L.len) return j-1;

else return -1;

}

elemtype sequenlist :: next(sequenlist L,elemtype x)

{

int j=1;

while((j<=L.len)&&(L.a[j]!=x))

j++;

if(j<L.len) return j+1;

else return -1;

}

void sequenlist ::Dele(sequenlist &L , int i)

{ int j;

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

cout<<" position is not correct!"<<endl;

else {

for(j=i+1;j<=L.len;j++)

L.a[j-1]=L.a[j]; //元素前移

L.len--; //表长度减1

}

}

int sequenlist::Locate(sequenlist L , elemtype x)

{ int j=1;

while((j<=L.len)&&(L.a[j]!=x))

j++;

if(j<L.len) return j;

else return -1;

}

int sequenlist :: length(sequenlist L )

{

return L.len;

}

void sequenlist ::setnull(sequenlist &L )

{

L.len=0;

}

void sequenlist ::prin(sequenlist L)//显示出线性表里的元素

{

cout<<"线性表的元素为：";

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

cout<<L.a[i]<<' ';

cout<<endl;

}

void sequenlist ::Insert(sequenlist &L,elemtype x,int i)

{

int j;

if(L.len==maxsize-1)

cout<<"overflow"<<endl;

else if ((i<1)||(i>L.len+1))

cout<<"position is not correct!"<<endl;

else

{

for(j=L.len;j>=i;j--)

L.a[j+1]=L.a[j]; //元素后移

L.a[i]=x; //插入元素

L.len++; //表长度增加1

}

}

void main()

{

int i;

sequenlist rj1;

rj1.setnull(rj1);

int t;

cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*输入数字1为创建顺序表：\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;

cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*输入数字2为显示线性表的长度：\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;

cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*输入数字3为插入一个元素到线性表：\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;

cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*输入数字4为删除一个元素到线性表：\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;

cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*输入数字5为查找线性表中某个元素的位置：\*\*\*\*\*\*\*"<<endl;

cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*输入数字6为取出一个位置的元素：\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;

cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*输入数字7为查看某个元素的前驱：\*\*\*\*\*\*\*"<<endl;

cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*输入数字8为查看某个元素的前驱：\*\*\*\*\*\*\*"<<endl;

cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*输入数字9为置表空\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;

cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*输入数字0为结束操作\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;

cout<<"请输入操作数：";

cin>>t;

while(t!=0)

{

switch(t)

{case 1:

{

int x;

cout<<"插入几条数据: "<<endl;

cin>>x;

for( i=1;i<=x;i++)//为数据赋元素；

{

cout<<"请输入第: "<<i<<"个元素：";

cin>>rj1.a[i];

rj1.len++;

cout<<endl;

}

rj1.prin(rj1);

}//输出线性表

break;

case 2:

{ cout<<"线性表的长度为: "<<rj1.length(rj1) <<endl;}// 3、求顺序表的长度

break;

case 3:

{ int x,j;

cout<<"请输入插入元素的值: ";

cin>>x;

cout<<"请输入插入的位置: ";

cin>>j;

rj1.Insert(rj1,x,j);

rj1.prin(rj1);}

break;

case 4:

{ int j;

cout<<"请输入删除的位置: ";

cin>>j;

rj1.Dele(rj1,j);

rj1.prin(rj1);}

break;

case 5:

{int x;

cout<<"请输入要查找的元素的值: ";

cin>>x;

cout<<"待查找的元素的位置是: "<<rj1.Locate(rj1, x)<<endl; }

break;

case 6:

{ int x;

cout<<"请输入取出元素的位置: ";

cin>>x;

cout<<"元素是: "<<rj1.Get(rj1,x)<<endl;}

break;

case 7:

{ int x;

cout<<"请输入查看哪个元素的前驱: ";

cin>>x;

cout<<"此元素的前驱为: "<<rj1.prior(rj1,x)<<endl;}

break;

case 8:

{ int x;

cout<<"请输入查看哪个元素的后继：";

cin>>x;

cout<<"此元素的后继为: "<<rj1.next(rj1,x)<<endl;}

break;

case 9:

{

cout<<"置表空---";

rj1.setnull(rj1);}

break;

default:cout<<"您输入的位置有误！";

}

cout<<"请输入数字：";

cin>>t;

}

}

}

**【实验心得】**

**深刻明白了顺序链表的增删改查等一系列操作，懂得了搭配链表模板，也明白了各个算法中的引用&的用处，进一步体会了函数调用等面向对象的思想，通过所学知识逐步完善主函数，首先是顺序表方法的调用问题得以解决，其次while循环和switch的配套使用，使得程序更加完整，逻辑更加严密**