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 链表的基本操作 实验日期 2017.10.16

同组人员 指导老师 汪红霞 得分

**实 验 项 目 二**

**【实验名称】**链表的基本操作

**【实验类型】**基础性实验

**【实验目的与要求】**

实验目的  
1． 掌握单链表的基本操作：建立、插入、删除、查找等运算。  
 实验要求  
1． 认真阅读和掌握本实验的程序。  
2． 上机运行本程序。  
3． 保存和打印出程序的运行结果，并结合程序进行分析。  
4． 按照你对单链表的操作需要，重新改写主程序并运行，打印出文件清单和运行结果

**【实验主要仪器设备及台套数】**

PC机,每人一台

**【实验学时】**  2学时

**【实验内容】**

1. 实现单链表存储的各种操作（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>

typedef int elemtype;

class link

{

public:

elemtype data; //元素类型

link \*next; //指针类型，存放下一个元素的地址

public:

void length(link \*head); //求链表的长度

void Insert(link \*head , elemtype x , elemtype y); //将元素x插入到第i个位置

void link::dele(link \*head , elemtype x); //删除第i个位置的元素

void setnull( );//置空表

link \*Locate(link \*head ,elemtype x); //定位，查找元素x的位置

link \*get(link \*head , int i); //取第i个位置的元素

elemtype prior(elemtype x ); //求元素x的前驱

elemtype succ(elemtype x ); //求元素x的后继

link \*hcreat1(int n );

link \*hcreat2(int n );//头插法建立单链表

link \*rcreat1( int n);

link \*rcreat2( int n);//尾插法建立单链表

void print(link \*head);

};

link \*link::rcreat1(int n )

{

link \*s,\*r,\*p;

int i;

p=NULL;

for(i=1;i<=n;i++)

{

s=new link;

cin>>s->data;

if(p==NULL) p=s;

else r->next=s;

r=s;

}

r->next=NULL;

return p;

}

link \*link::rcreat2(int n )

{

link \*s,\*r,\*p;

int i;

p=r=new link;

p->next=NULL;

cout<<"创建单链表： ";

for(i=1;i<=n;i++)

{

s=new link;

cin>>s->data;

r->next=s;

r=s;

}

r->next=NULL;

return p;

}

link \*link::hcreat1(int n )

{

link \*s,\*p;

int i;

p=new link;

p->next=NULL;

for(i=1;i<=n;i++)

{

s=new link;

cin>>s->data;

s->next=p->next;

p->next=s;

}

return p;

}

link \*link::hcreat2(int n )

{

link \*s,\*p;

int i;

p=new link;

p->next=NULL;

for(i=1;i<=n;i++)

{

s=new link;

cin>>s->data;

s->next=p->next;

p->next=s;

}

return p;

}

void link::length(link \*head )

{

link \*p;

p=head->next;

int count=1;

while(p->next!=NULL)

{

p=p->next;

count++;

}

cout<<"单链表的长度为:"<<count<<endl;

}

link \*link::Locate(link \*head ,elemtype x)

{

link \*p;

p=head->next;

if(p!=NULL){

while(p->data!=x)

p=p->next;

}

else

{

cout<<"次元素不存在"<<endl;

return NULL;

}

return p;

}

link \*link::get(link \*head , int i)

{

int j;

link \*p;

j=1;

p=head->next;

if(p!=NULL){

while(j<i)

{

j++;

p=p->next;

}

}

else

{

cout<<"次元素不存在"<<endl;

return NULL;

}

return p;

}

void link::Insert(link \*head , elemtype x , elemtype y)

{

link \*p,\*s;

s=new link;

s->data=x;

if(head->next==NULL) //链表为空

{

head->next=s;

s->next=NULL;

}

p=Locate(head,y) ; //调用查找算法

if(p==NULL)

cout<<"插入位置非法";

else

{ s->next=p->next;

p->next=s;

}

}

void link::dele(link \*head , elemtype x)

{

link \*p,\*q;

q=head;

p=head->next;

while((p!=NULL)&&(p->data!=x))

{

q=p;

p=p->next;

}

if(p==NULL) cout<<"要删除的结点不存在";

else

{

q->next=p->next;

delete(p);

}

}

void link::print(link \*head)

{

link \*p;

p=head->next;

while(p->next!=NULL)

{

cout<<p->data<<"->"; //输出表中非最后一个元素

p=p->next;

}

cout<<p->data; //输出表中最后一个元素

cout<<endl;

}

void main()

{

link rj1,\*p,\*p1;

//int m,x,y;

//cout<<"请输入链表的节点数：";

// cin>>m;

//p=rj1.rcreat2(m);

//p=rj1.hcreat2(m);

//rj1.print(p);

//cout<<"请输入想要查找的元素： ";

//cin>>m;

// cout<<"请输入插入元素的值： ";

// cin>>m;

// cout<<"请输入待插入的元素： ";

// cin>>m;

// rj1.Insert(p,m,m);

// rj1.print(p);

// cout<<"请输入删除元素的值： ";

// cin>>m;

// rj1.dele(p,m);

// rj1.print(p);

cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*输入数字1为创建单链表：\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<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9为置表空\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;

cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*输入数字7为查看单链表\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;

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

cout<<"请输入操作数：";

int t;

cin>>t;

while(t!=0)

{

switch(t)

{case 1:

int m;

cout<<"请输入链表的节点数：";

cin>>m;

p=rj1.rcreat2(m);

rj1.print(p);

break;

case 2:

{ rj1.length(p);}// 求单链表表的长度

break;

case 3:

{

int m,n;

cout<<"请输入插入元素的值： ";

cin>>m;

cout<<"请输入待插入的元素： ";

cin>>m;

rj1.Insert(p,m,n);

rj1.print(p);}

break;

case 4:

{ int j;

cout<<"请输入删除的位置: ";

cin>>j;

rj1.dele(p,j);}

break;

case 5:

{

cout<<"请输入查找某一元素的位置： ";

cin>>m;

p1=rj1.Locate(p,m);

cout<<p1->next<<endl;}

break;

case 6:

{ cout<<"请输入查找位置的元素： ";

cin>>m;

p1=rj1.get(p,m);

cout<<p1->data<<endl;}

break;

case 7:

{

cout<<"查看结果: "<<endl;

rj1.print(p);

}

break;

default:return ;

}

cout<<"请输入数字：";

cin>>t;

}

}

**【实验心得】**

**本次实验实现了，单链表的增删改查，并在main方法里优化了程序，提高程序的健壮性，也深刻明白了单链表的增删改查等一系列操作，懂得了搭配单链表模板，也明白了各个算法中的指针的用处，进一步体会了函数调用等面向对象的思想，体会到单链表与顺序表的优缺点与异同，受益匪浅。**