计算机与信息工程学院实验报告

••••••••••••••••••••••••••••••••• 密 ••••••••••••••••••••••••••••••••• 封 ••••••••••••••••••••••••••••••••• 线 •••••••••••••••••••••••••••••••••

姓名：朱龙康 学号：1824120002 专业：计算机科学与技术\_ 年级：\_2018级

课程：数据结构 主讲教师：\_袁彩虹\_ 辅导教师：\_\_袁彩虹\_\_

实验时间：\_\_2020\_年 \_3\_月 \_\_20\_日 \_上\_午\_8\_时至\_10\_时，实验地点\_\_\_\_\_\_\_\_

实验题目： 单链表的操作

实验目的： 1）掌握线性表的链式存储结构。2）掌握单链表及其基本的实现

实验环境（硬件和软件） CodeBlocks

实验内容：

（1）编程实现单链表的基本操作：建立单链表，查找单链表，插入单链表，删除单链表；

（2）采用单链表结构编程实现：两个有序单链表的归并运算。

实验步骤：

1）**建立单链表，查找单链表，插入单链表，删除单链表**

//1.初始化

bool InitList(LinkList &L){ //指针类型的引用 必须要加& 否则没法改变指针L

L=new LNode;

L->next=NULL;

return true;

}

//2.后插法创建长度为n的链表 需要尾指针

void CreateList\_R(LinkList &L,int n){

InitList(L);//初始化L

LinkList r=L;//创建尾指针

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

LinkList p=new LNode;//生成新节点

cin>>p->data;//输入数据

p->next=NULL;//next值空

r->next=p;//尾插

r=p;//尾指针后移

}

}

//3.头插法创建长度为n的链表 需要头指针

void CreateList\_H(LinkList &L,int n){

InitList(L);//初始化L

LinkList r=L;//创建头指针

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

LinkList p=new LNode;//生成新节点

cin>>p->data;//输入数据

p->next=r->next;//头差

r->next=p;

}

}

//4.查找

LNode \*SearchE(LinkList l,int e){

LinkList p=l.next;//初始化指针p

while(p&&p->data!=e){//没有找到e

p=p->next;//指针后移

}

return p;//返回已指向的带e节点的指针p

}

//5.插入

bool Insert(LinkList &l,int i,int e){//在链表l的第i个位置插入元素e

LinkList p=l;//初始化p

k=0;

while(p&&k<i-1){//p不为空，一直遍历到第i-1个位置

p=p->next;k++;

}

if(!p&&i<1){return false;}//输入的i不和法

LinkList n=new LNode;//生成新节点

n->data=e;

n->next=p->next;//新节点next指向第i个位置

p->next=n;//第i-1个位置next指向新节点

return true;

}

//6.删除

bool Insert(LinkList &l,int i){//删除链表第i个节点

LinkList p=l;//初始化p

k=0;

while(p->next&&k<i-1){//p不为空，一直遍历到第i-1个位置

p=p->next;k++;

}

if(!(p->next)&&i<1){return false;}//输入的i不和法

LinkList n=p->next;//生成新指针 临时保存要删除的节点

p->next=n->next;//第i-1个位置的next指针指向第i个位置的后一个位置

delete n;//释放第i个节点

return true;

}

**2）两个有序单链表的归并运算**

//有序链表归并

//假设按升序排列

void AandB(LinkList&la,LinkList&lb,LinkList&lc){

LinkList pa=la->next,pb=lb->next;//pa，pb指向la，lc的首元节点

lc=la;//lc等于la的头结点

LinkList pc=lc;//指针pc指向lc的表头

while(la&&lb){//当指针均为到达两个链表尾部

if(pa->data<=pb->data){//如果pa元素更小

pc->next=pa;//pc的next指向pa

pc=pa;//pc指针后移

pa=pa->next;//pa指针后移

}

else{//与前相反

pc->next=pb;

pc=pb;

pb=pb->next;

}

}

if(pa)pc->next=pa;//如果pa没有遍历完 则在pc位置接上pa

else pc->next=pb;//相反

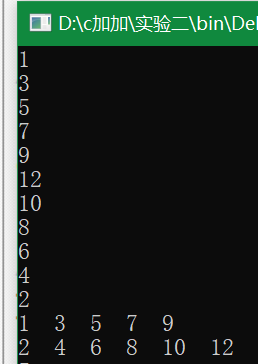
delete lb;//释放lb的表头

}

实验数据记录：

1.创建两个单链表L1和L2 数据域分别为1,3,5,7,9 2,4,6,8,10,12

打印两个单链表的数据域



2.查找数据为5的节点



3.删除L2的第6个节点



4.L1和L2归并



问题讨论：