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

链表的实验报告

**一、实验目的**：了解线性表的链式存储结构，熟悉掌握链式存储结构的描述方法，链表是本章的重点与难点，熟练掌握链表的基本操作查找、插入等，能在实际应用中选择适当的链表结构。

**二、实验要求**：

1）定义链表结构

2）写函数创建链表

3）任选实验内容中1、2两题之一，写函数完成要求。

4）写主函数，调试运行程序，显示结果。

**三、实验内容：**

1、 （作业2.19）已知线性表中的元素以值递增有序排列，并以单链表作存储结构。试写一高效算法，删除表中所有值大于mink且小于maxk的元素（若表中存在这样的元素）同时释放被删结点空间，并分析你的算法的时间复杂度（注意：mink和maxk是给定的两个参变量，他们的值可以和表中的元素相同，也可以不同）。

2、 1） 逆序建立单链表

2） 遍历单链表(输出单链表每个元素的值)

3） 在单链表第5个元素前插入一个值为999的元素.

4） 删除单链表第5个元素.

[输入要求]

输入n个字符串（长度小于6），建立含有n个结点的单链表，输入的字符串不重复。

[测试数据]

例如：

建立时输入：

n=5

red green yellow blue black

插入时输入：orange (在第5个结点之后插入)

删除时输入：blue

**四、算法思想：**

线性表的链式存储结构的特点是用一组任意的存储单元存储线性表的数据元素（这组存储单元可以是连续的，也可以是不连续的）。所以对数据元素a来说，不仅要存储数据元素的a的存储映像，还要存储其直接后继的信息；通过对这种链表结构各种基本操作的实现来完成实验内容。

**五、主要函数的实现：**

**//----常量头文件---**

**#ifndef \_CONSTANT\_H**

**#define \_CONSTANT\_H**

**typedef int ElemType;**

**typedef int Status;**

**#define TRUE 1**

**#define FALSE 0**

**#define OK 1**

**#define ERROR 0**

**#define INFEASIBLE -1**

**#define OVERFLOW -2**

**#endif**

//---链表头文件---

#ifndef \_LinkList\_H

#define \_LinkList\_H

#include<stdlib.h>

#include "constant.h"

typedef struct LNode{

ElemType data;

struct LNode \*next;

}LNode,\*LinkList;

Status InitList(LinkList L){

int n;

LinkList p,q;

p=L;

printf("请输入n和n个数据元素:\n");

scanf("%d",&n);

while(n--)

{

q=(LinkList)malloc(sizeof(LNode));

scanf("%d",&q->data);

q->next=NULL;

p->next=q;

p=q;

}

return OK;

}

Status PrintfList(LinkList L){

int num=0;

LinkList p;

p=L->next;

while(p)

{

num++;

printf("%d ",p->data);

p=p->next;

}

printf("\n长度为%d:\n",num);

return OK;

}

void Find(LinkList L,int x)//查找值为x的直接前驱结点q

{

LinkList p;

p=L;

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

p=p->next;

if(p->next)

printf("%d的前驱结点为:%d\n\n",x,p->data);

else

printf("没找到!!\n\n");

}

void Delete(LinkList L,int x)//删除值为x的结点

{

LinkList p,q;

p=L;

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

p=p->next;

if(p->next)

{

q=p->next;

p->next=q->next;

free(q);

printf("删除成功!!\n\n");

}

else

printf("链表中没有%d\n\n",x);

}

Status Delete\_Between(LinkList L){

int maxk,mink;

LinkList p,q,s;

printf("请输入mink,maxk:\n");

scanf("%d %d",&mink,&maxk);

p=L;

while(p->next && p->next->data<=mink)

p=p->next;

s=p->next;

while(s && s->data<maxk)

{

q=s;

s=s->next;

free(q);

}

p->next=s;

printf("删除成功\n\n");

return OK;

}

void Insert(LinkList L,LinkList p)//在升序链表插入值为X的结点，使仍然有序

{

LinkList s;

s=L;

while(s->next && s->next->data < p->data)

s=s->next;

p->next=s->next;

s->next=p;

}

Status Sort(LinkList L)//按升序排列

{

LinkList s,r;

s=L->next;

L->next=NULL;

while(s)

{

r=s;

s=s->next;

r->next=NULL;

Insert(L,r);

}

return OK;

}

#endif

//--主函数---

#include <stdio.h>

#include <stdlib.h>

#include"LinkList.h"

int main(int argc, char \*argv[])

{

int n;

LinkList L;

InitList(L);

PrintfList(L);

Delete\_Between(L);

PrintfList(L);

system("PAUSE");

return 0;

}

**六、测试结果：**

****

**七、实验总结：**

**1.逻辑不够清晰，如果再多考虑一下使用的人就更好了；**

**2.对常量的使用还是不够熟练；**

**3.运行结束后按几个回车键后程序还是会出问题；**