《数据结构》课程实验报告

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 专业名称 | 计算机 | 年级 | 2017 | 班级 | 2 |
| 学生姓名 | 王汝芸 | 学号 | 201711010202 | 指导教师 | 郑志华 |
| 实验题目 | 实验2：单链表的创建及操作 | | | 提交时间 | 2018．3.18 |

一、实验目的和要求

例如：（1）熟悉C语言的上机环境，进一步掌握C语言的结构特点。

（2）掌握单链表的顺序存储结构的定义及基本运算

二、实验环境

Visual Studio2017

三、实验内容及实施

采用较为简单的线性表；宏定义N为数组长度，方便后期修改；快速排序法；\*没按顺序实现问题

**实验1：顺序表的建立及运算**

要求：

1、创建一个带头结点的单链表（头指针为head），且遍历此链表（输出链表中各结点的值）；

2、查找单链表中的第i个结点，并输出结点元素的值；

3、在单链表中的第i个结点前插入一个结点值为e的正整数(从外部输入);

4、删除单链表中的第j个结点；

\*5、将单链表中的各结点就地逆序（不允许另建一个链表）；

**【源程序】**

/\* 数据结构实验二

——单链表的创建及操作

\*/

/\* 1、创建一个带头结点的单链表（头指针为head），且遍历此链表（输出链表中各结点的值）；√

2、查找单链表中的第i个结点，并输出结点元素的值；√

3、在单链表中的第i个结点前插入一个结点值为e的正整数(从外部输入);√

4、删除单链表中的第j个结点；√

\*5、将单链表中的各结点就地逆序（不允许另建一个链表）；

\*/

#include<stdio.h>

#include<malloc.h>

struct Linklist

{

int data;

Linklist \*next;

};

void Createlist(Linklist \*&L,int n)//n为链节个数,L为头结点

{

int i;

Linklist\*p;//指向节点

L = (Linklist\*)malloc(sizeof(Linklist));

L->next = NULL;//初始化头结点

//从头结点后插入新节点

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

{

p = (Linklist\*)malloc(sizeof(Linklist));

printf("请输入数字\n");

scanf\_s("%d", &p->data);//输入

p->next = L->next;

L->next = p;

}

}

int main()

{

//头结点

Linklist \*head;

printf("请输入5个数字：\n");

//创建5个节点的链表

Createlist(head, 5);

Linklist\*p = head->next;

//输出

printf("输出：");

while (p != NULL)

{

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

p = p->next;

}

putchar('\n');

int k;

printf("查找第k个元素的值，请输入k:\n");

scanf\_s("%d", &k);

int i;

p = head->next;//初始化p

//输出第k个元素的值

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

{

if (i == k)

{

printf("第%d个元素为：%d\n",k, p->data);

break;

}

p = p->next;

}

//在k前插入e

printf("在k前插入e，请输入k：");

scanf\_s("%d", &k);

int e;

printf("请输入插入的数值e：");

scanf\_s("%d", &e);

p = head->next;//初始化p

Linklist\*q;

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

{

if (i == k-1)

{

q = (Linklist \*) malloc(sizeof(Linklist));

q->data = e;

q->next = p->next;

p->next = q;

break;

}

p = p->next;

}

p = head->next;

putchar('\n');

//输出链表

while (p != NULL)

{

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

p = p->next;

}

putchar('\n');

//删除链表中第j个节点

int j;

printf("删除链表中第j个节点，请输入j：");

scanf\_s("%d", &j);

p = head->next;

q = head->next;

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

{

if (i == j-1)

{

q=p->next;

p->next = q->next;

free(q);

break;

}

p = p->next;

}

p = head->next;

//输出链表

while (p != NULL)

{

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

p = p->next;

}

putchar('\n');

//就地逆序

p = head->next;

head->next = NULL;

while (p != NULL)

{

q = p->next;

p->next = head->next;

head->next = p;

p = q;

}

printf("逆序后链表为：\n");

p = head->next;

//输出链表

while (p != NULL)

{

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

p = p->next;

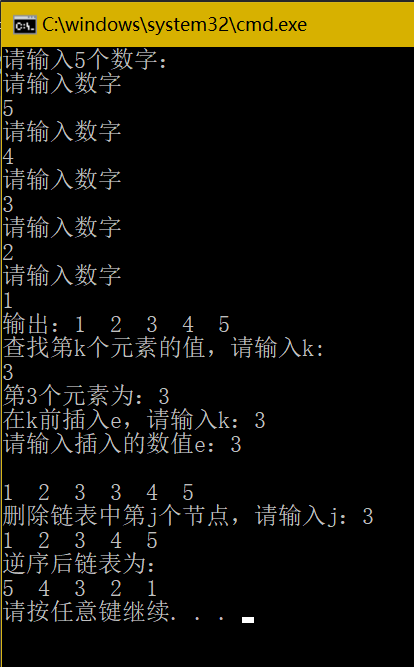
}

putchar('\n');

return 0;

}

**四、实验结果 (程序的执行结果)**



**五、实验讨论（可选）**

无