

数据结构链表操作

编写一个程序，要求：从终端输入一组整数（大于10个数），以0作为结束标志，将这一组整数存放在一个链表中（结束标志0不包括在内），打印出该链表中值。然后删除该链表的第5个元素，打印出删除后的结果。最后在内存中释放掉该链表。

```
#include<stdio.h>
#include <stdlib.h>
typedef int ElemType;
typedef struct node{
    ElemType data;           /*数据域*/
    struct node *next;       /*指针域*/
}LNode,*LinkList;
LinkList GreatLinkList(int n)           /*创建一个链表，包含n个结点*/
{
    LinkList p,r,list=NULL;
    ElemType e;
    int i;
    for(i=1; i<=n; i++)
    {
        //尾插法
        scanf("%d", &e);           /*输入结点内容*/
        p=(LinkList)malloc(sizeof(LNode));           /*为新建的结点开辟内存空间*/
        p->data=e;           /*元素赋值*/
        p->next=NULL;
        if(!list)
            list=p;           /*赋值链表头指针*/
        else
            r->next=p;           /*将结点连入链表*/
        r=p;
    }
    return list;           /*返回链表头指针*/
}
void insertList(LinkList *list, LinkList q, ElemType e)           /*向链表中插入结点*/
{
    LinkList p;
    p=(LinkList)malloc(sizeof(LNode));           /*为新建结点开辟内存空间*/
    p->data=e;
    if(!*list)
    {
        *list=p;           /*赋值链表头指针*/
        p->next=NULL;
    }
    else           /*将结点连入链表*/
    {
        p->next=q->next;
        q->next=p;
    }
}
void delLink(LinkList *list, LinkList q)           /*删除链表的某结点*/
{
    LinkList r;
    if(&q==list)           /*如果删除第一个结点*/
    {
        *list=q->next;
        free(q);
    }
}
```

```

}
else                                /*删除其他结点*/
{
    for(r=*list; r->next!=q; r=r->next);
    if(r->next!=NULL)
    {
        r->next=q->next;
        free(q);
    }
}
}
void destroyLinkList(LinkList *list)    /*销毁一个链表*/
{
    LinkList p, q;
    p=*list;
    while(p)                            /*循环释放掉每一个链表*/
    {
        q=p->next;
        free(p);
        p=q;
    }
    *list=NULL;
}
main()
{
    int e, i;
    LinkList l, q;
    q=l=GreatLinkList(1);              /*创建一个链表结点，q和l指向该结点*/
    scanf("%d", &e);
    while(e)                            /*循环地输入数据，同时插入新生成的结点*/
    {
        insertList(&l, q, e);
        q=q->next;
        scanf("%d", &e);
    } //因为前面调用了GreatLinkList函数，而该函数用尾插法建立链表的，p->next=NULL;保证了最后一个结点指向NULL。
    q=l;                                //前面指向第一个结点，所以此处保证q回到第一个结点处。
    printf("The content of the linklist\n");
    while(q)                            /*输出链表中的内容*/
    {
        printf("%d ", q->data);
        q=q->next;
    }
    q=l;
    printf("\nDelete the fifth element\n"); //删除第五元素
    for(i=0; i<4; i++)                  //将指针q指向链表第五个元素
    {
        q=q->next;
    }
    delLink(&l, q);                      //删除q所指的结点
    q=l;
    while(q)                            //打印删除后的结果
    {
        printf("%d ", q->data);
        q=q->next;
    }
    destroyLinkList(&l);                 //释放掉该链表
}

```

```
printf("\n");  
}
```