

咸鱼注：大家可以在自己电脑上新建一个 C++ 项目，并把这段代码复制到 main 函数前边，然后在你的 main 函数里调用 test_LinkList() 这个函数。

提示：用电脑网页版看，体验更好一些

```
// 个人微博：@王道咸鱼学长-计算机考研
// 这段代码实现了单链表的查找：双指针法（较常用）

#ifndef CODE_LINKLIST_H
#define CODE_LINKLIST_H

#include <stdio.h>
#include <stdlib.h>

//定义单链表结点
typedef struct LNode{           //定义单链表结点类型
    int data;                   //每个节点存放一个数据元素
    struct LNode *next;        //指针指向下一个节点
}LNode, *LinkList;

//初始化一个单链表（带头结点）
bool InitList(LinkList &L) {
    L = (LNode *) malloc(sizeof(LNode)); //分配一个头结点
    if (L==NULL)                          //内存不足，分配失败
        return false;
    L->next = NULL;                        //头结点之后暂时还没有节点
    return true;
}

//输出链表相关的所有信息(带头结点)
void PrintList(LinkList L){
    LNode *p = L->next; //指向第1个结点
    int len = 0;         //统计链表的长度
    printf("L: 头结点 -> ");
    while (p!=NULL){
        len++;           //单链表统计长度+1
        printf("%d -> ", p->data);
        p = p->next;
    }
    printf("NULL\n链表长度 = %d\n", len);
}

//双指针法，通过一次遍历找到中间结点
LNode * findMidNode(LinkList head){
    LNode * p=head;      //p指针，每次往后走一步
    LNode * q=head;      //q指针，每次往后走两步，当q指向最后一个结点时，p刚好指向中间那个结点
    while (q->next!=NULL){
        q = q->next;      //q往后走一步
        if (q->next!=NULL)
            q = q->next;   //q再往后走一步
        p=p->next;        //p往后走一步
    }

    if(p==head)
        printf("一个空链表，让我找中间结点？你一定是在逗我~");
    else
        printf("中间结点的值为: %d\n", p->data);
    return p; //返回p所指向的结点
}

//双指针法，通过一次遍历找到倒数第k的结点
LNode * findDaoShuKNode(LinkList head, int k){
    LNode * p=head;      //p指针，等q指针走到第k个结点时，p指针再出发
    LNode * q=head;      //q指针，每次往前走一步

    int count=0;
    bool chong = false;   //p指针要不要冲？
    while(q->next!=NULL){
        q = q->next;       //q往后走一步
        count++;           //计数+1
        if (count==k)
            chong = true;  //q走到第k个结点时，p指针开冲

        if(chong)
            p = p->next;
    }

    if(p==head)
        printf("这个链表长度小于%d，并不存在倒数第k个结点\n", k);
    else
```

```

        printf("倒数第%d个结点的值为: %d\n", k, p->data);
    return p;    //返回p所指向的结点
}

#define N 20
//【测试代码】：建立一个单链表，并尝试用双指针法找中间结点、倒数第K个结点
void test_LinkList() {
    LinkList head;    //声明一个单链表
    InitList(head);    //初始化，新建一个头结点
    //插入一些测试数据，从1~N
    LNode * p = head;
    for (int i=1; i<=N; i++){
        p->next = (LNode *) malloc(sizeof(LNode));
        p = p->next;
        p->data = i;
        p->next = NULL;
    }
    PrintList(head);    //打印出来看一眼

    findMidNode(head);

    findDaoShuKNode(head, 3);
}

#endif //CODE_LINKLIST_H

```