#include<stdio.h>

#include<stdlib.h>

typedef struct node

{

    int k;

    struct node \*next;

}ListNode;

ListNode \*CreateNode(int arr[],int len)

{

    ListNode \*head =(ListNode\*)malloc(sizeof(ListNode));

    //初始化

    head->next = NULL;

    ListNode \*end = head;//尾指针初始化

    for (int i = 0; i < len; i++)

    {

        ListNode \*p = (ListNode\*)malloc(sizeof(ListNode));//数组元素建立节点

        p->k = arr[i];

        end->next = p;//将节点p插入到终端

        end = p;

    }

    end->next = NULL;//链表创建成功，终端指针域置空

    //返回新建节点地址

    return head;

}

//头插法

void InsertNodeToHead(ListNode \*head,int k)

{

    ListNode \*new\_Node = malloc(sizeof(ListNode));

    //初始化

    new\_Node->k = k;

    new\_Node->next= NULL;

    //插入节点

    new\_Node->next =head->next;

    head->next = new\_Node;

}

//双指针查找单链表倒数第k个节点

ListNode \*kthToLast(ListNode \*head,int k)

{

    ListNode \*p = head;

    ListNode \*per = head;

    for (int i = 0; i < k; i++)

    {

        per = per->next;

    }

    while (per!= NULL)

    {

        p = p->next;

        per = per->next;

    }

    return  p;

}

void printfNode(ListNode \*tmp)

{

    if (tmp == NULL) return ;

    ListNode \*p=tmp->next;

    while (p)

    {

        printf("%d\t",p->k);

        p = p->next;

    }

    printf("\n");

}

int main(void)

{

   int arr[] = {1,2,3,4,5,6,7};

   int len = sizeof(arr)/sizeof(int);

    ListNode \*head =CreateNode(arr,len);

    printfNode(head);

    printf("------------\n");

    //查找链表倒数第k个节点

    ListNode\*temp = kthToLast(head,4);

    printf("找到了k= %d\n",temp->k);

    return 0;

}