



链滴

# 查找单链表倒数第 $k$ 个元素

作者: [undersky](#)

原文链接: <https://ld246.com/article/1479175367591>

来源网站: 链滴

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

查找单链表倒数第m个结点，要求时间复杂度为O(n).(提示，使用双指针)

---

解题思路：

常规思路为先获取链表的长度N，然后返回N-k+1位置处的结点即可。但是中需要遍历两次链表。

我们使用另一种算法，设定两个指针p1,p2.将这两个指针都向第一个结点，让p1先走k步，然后两个针一起向后移动，当p1到达最后一个结点时，p2指针刚好指向链表的倒数第k个结点。

关键算法：

```
int getK(Node *link, int k)
{
    Node *p1, *p2;
    int i;
    if (link->next == NULL || k == 0)
        return -1;
    p1 = link->next;
    for (i = 1; i < k; i++)
    {
        if (p1->next == NULL)
            return -1;
        p1 = p1->next;
    }
    p2 = link->next;
    while (p1->next != NULL)
    {
        p1 = p1->next;
        p2 = p2->next;
    }
    return p2->number;
}
```

---

完整程序如下：

```
/******
Problem: 查找单链表倒数第N个元素，要求复杂度O(n)
User: xiaoming
Language: C
*****/

#include
#include

typedef struct node
{
    int number;
    struct node * next;
} Node;

int getK(Node *link, int k);
```

```

int main()
{
    int n, k, i, temp;
    int flag;
    while (scanf("%d %d", &n, &k) != EOF && n >= 0 && n <= 1000 && k >= 0 && k <= 1000)

    {
        Node *link = (Node *)malloc(sizeof(Node));
        link->next = NULL;
        flag = 0;
        Node *tail;
        tail = link;
        for (i = 0; i < n; i++)
        {
            //scanf("%d", &temp);
            Node *n = (Node *)malloc(sizeof(Node));
            n->next = tail->next;
            tail->next = n;
            tail = tail->next;
            n->number = i + 1;
        }
        int numberK = getK(link, k);
        numberK == -1 ? printf("NULL\n") : printf("%d\n", numberK);
    }
    return 0;
}

int getK(Node *link, int k)
{
    Node *p1, *p2;
    int i;
    if (link->next == NULL || k == 0)
        return -1;
    p1 = link->next;
    for (i = 1; i < k; i++)
    {
        if (p1->next == NULL)
            return -1;
        p1 = p1->next;
    }
    p2 = link->next;
    while (p1->next != NULL)
    {
        p1 = p1->next;
        p2 = p2->next;
    }
    return p2->number;
}

```