

# 习题 0810

## 说明

本次习题共 3 道编程题，综合题需要在周日下午一点前提交，前两题需要在习题时间结束前完成。

## 01 环形链表的长度

APD 写了一个单向环形链表，但他忘记自己曾经往里面加入了多少个元素了（而且他还往里面加了点内容相同的元素），你能帮他数数吗？

### 输入

环形链表里其中一个元素的地址（指针）

### 输出

环形链表中元素的个数

### 样例输入

```
1 1(输入) -> 2 -> 3 -> 4 -> 1 -> 5 -> 1(输入) -> 2 -> 3 -> ...
```

### 样例输出

```
1 6
```

### 数据说明

- 链表结构体内容如下：

```
1 typedef struct LinkedList {
2     int val;
3     struct LinkedList *next;
4 }RLL;
```

- 链表的长度小于 `int` 的上限

### 提示

- 遍历
- 想想链表节点的唯一标志是什么？

## 02 判断链表是否有效

为了防止链表内容被偷偷篡改，某公司人为地规定了链表的一种验证机制如下：

- 链表的长度小于 31
- 链表内的数字小于 1000
- 链表的首位（设下标为 0）是验证位置，内部存有一个验证数字
- 从链表下标为 1 开始的位置起，将各位数字乘以素数表中对应位置的素数（第一位对应 2，第二位对应 3，第三位对应 5，以此类推）并加和，最后对 1009 求模。如果得到的数字等于链表的首位，则称这个链表是有效的，反之则这个链表是无效的。

现在给你一个链表，你能不能判断这个链表是否是有效的？

【附】前 30 个素数是：

```
1  2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89 97 101 103 107
   109 113
```

## 输入

链表头节点的指针

## 输出

如果链表是有效的，则输出 1，反之则输出 0

### 样例输入 1

```
1  51 -> 1 -> 2 -> 3 -> 4
```

### 样例输出 1

```
1  1
```

解释：  $51 = (1*2 + 2*3 + 3*5 + 4*7) \% 1009$

### 样例输入 2

```
1  233 -> 1000 -> 0 -> 0 -> -1 -> 2
```

### 样例输出 2

```
1  0
```

解释：  $233 != (1000*2 + 0*3 + 0*5 + (-1)*7 + 2*11) \% 1009 = 1006$

## 数据说明

- 链表结构体内容如下：

```
1  typedef struct LinkedList {
2      int val;
3      struct LinkedList *next;
4  }LL;
```

## 03 强强进制转换器

APD 造了个进制转换器，可以把字符串从 2 到 16 进制里的任意一种进制转换为另一种进制，你也想整一个

## 输入

字符串 `s` 和这个数组原本的进制 `n0`，需要转换到的进制 `n1`

## 输出

将 `s` 以 `n0` 进制解读，并且转化为 `n1` 进制的字符串数组返回（字母均使用小写字母）

### 样例输入 1

```
1  "233"
2  4
3  2
```

### 样例输出 1

```
1  101111
```

### 样例输入 2

```
1  "2333"
2  8
3  16
```

### 样例输出 2

```
1  4db
```

说明：  $2333(8) = 1243(10) = 4db(16)$

## 数据说明

- `s` 在 `n0` 进制下的大小在 `int` 范围内

## 提示

- 可以使用预先设定好的数值 - 字符对应表简化操作