

#D、身份证校验

题目描述

一个合法的身份证号码由 17 位地区、日期编号和顺序编号加 1 位校验码组成。校验码的计算规则如下：

首先对前 17 位数字加权求和，权重分配为：{7, 9, 10, 5, 8, 4, 2, 1, 6, 3, 7, 9, 10, 5, 8, 4, 2}；然后将计算的和对 11 取模得到值 Z ；最后按照以下关系对应 Z 值与校验码 M 的值：

Z:	0	1	2	3	4	5	6	7	8	9	10
M:	1	0	X	9	8	7	6	5	4	3	2

现在给定一些身份证号码，请你验证校验码的有效性，并输出有问题的号码。

输入格式

输入第一行给出正整数 $N(\leq 100)$ 是输入的身份证号码的个数。随后 N 行，每行给出 1 个 18 位身份证号码。

输出格式

按照输入的顺序每行输出 1 个有问题的身份证号码。这里并不检验前 17 位是否合理，只检查前 17 位是否全为数字且最后 1 位校验码计算准确。如果所有号码都正常，则输出 All passed。

输入样例 1

4
320124198808240056

```
. 12010X198901011234
. 110108196711301866
. 3707041988121601X
```

输出样例 1

```
. 12010X198901011234
. 110108196711301866
. 3707041988121601X
```

输入样例 2

```
. 2
. 320124198808240056
. 110108196711301862
```

输出样例 2

```
. All passed
```

解题思路

首先我们输入 N 。然后把表打好，将权值和对应的校验码分别用两个数组先写出来。初始化一切变量，数组之后再输入 N 组身份证号，边输边判断。首先算出乘权求和的值；应该用存身份证号码的数组的前 17 位依次乘上对应的权值，比如第一位对应权值 7，第二位对应权值 9，第三位对应权值 10 等等。

权值表																	
位数	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
权值	7	9	10	5	8	4	2	1	6	3	7	9	10	5	8	4	2

我们只需要依次用身份证号码的第 1, 2, 3, 4... 17 位来乘上对应的权，再将所有权值累加在一起就可以了。这里需要注意的是将字符转成整数。如果我没有记错的话，可以在 `a[j]` 前面加一个 `int`，把它强制转换为 `int`。就可以这样写：`(int) a[j]`。当然这里我用的是减去对应的 ASCII 码也就是 '0'，对应的 ASCII 是 48（如果没记错的话）。当然我们还需要判断这些身份证号码是否合法，是不是出现了除了数字之外的其他东西。所以我们需要再加一个判断。如果不是数字就直接输出这个身份证号，因为他一定有问题。最好加一个 `break`。这里我忘了。下面是求权值和身份证号码前 17 各位乘积之和的代码：

```
13 for (j = 0; j < 17 && a[j] >= '0' && a[j] <= '9'; j++){
14     if (isdigit(a[j]) == 0){
15         cout << a << endl;
16     }
17     sum += (a[j] - '0') * weight[j];
18 }
```

接着就只需要判断了。如果身份证号校验码不等于求出来的加权和 mod 11 的值就输出。反之将计数器加 1。如果最后计数器为 N，就说明这 N 组身份证号码全部 OK，输出 All passed。

至此，全题终。

参考：CSDN

网址：

https://blog.csdn.net/weixin_45344430/article/details/109548380?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522166461927816782425114999%2522%252C%2522scm%2522%253A%252220140713.130102334..%2522%257D&request_id=166461927816782425114999&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~all~top_positive~default-1-109548380-null-null.142^v51^control,201^v3^control_2&utm_term=%E5%8A%A0%E6%9D%83%E6%B1%82%E5%92%8C&spm=1018.2226.3001.4187