一篇文章带你搞懂mod 11-2算法

阅前提示：本文章仅供研究学习使用[1]

引语：mod 11-2算法目前广泛运用于各类加密场景，包括但不限于身份证最后一位的计算等。本文以身份证最后一位数字（校验码）的计算为例子，简单讲解mod 11-2算法。[2]

我国公民身份号码由十八位数字组成，大致可分割为以下几个部分[3]：

出生地省区代码	出生地城市代码	出生地县区代码	出生日期（年）	出生日期（月）	出生日期（日）	顺序码	校验码
44	09	02	1990	07	24	153	1
广东省	茂名市	茂南区				（奇数表男性，偶数表女性）	

那这个“1”是怎么来的呢？先看下面一组国标规定的加权因子[4]：

7 9 10 5 8 4 2 1 6 3 7 9 10 5 8 4 2

这个因子的计算方法[2]：

$$2^{18-i} \bigoplus 11$$

其中，**i**为各个号码的位数。

这组因子通常是固定的，我们可以直接利用它来计算。

身份号码前十七位分别对应这组因子的每一个数字，我们先做对应相乘。以上面表格的号码为例，将每一位数字分别与其对应的加权因子相乘，

例如：

$$4\times7, 4\times9, 0\times10, 9\times5...$$

得到数字：

$$a_1, a_2, a_3, a_4... （a_1 = 28, a_2 = 36...）$$

求和，得到数字**b = 319**

我们再将该数字套用到这样一个公式里[2]：

$$(12 - (b \bigoplus 11)) \bigoplus 11$$

最后算出来的数字即为校验码。

计算过程如下[2]：

$$\because b = 319,$$
$$\therefore b \bigoplus 11 = 319 \div 11\text{的余数} = 0$$
$$\therefore 12 - 0 = 12$$
$$\therefore 12 \div 11\text{的余数} = 1$$
$$\therefore \text{校验码为}1$$

用一个函数写下来就是[2]：

$$m = (12 - (\sum_{n=1}^{a_{17}} a_n \bigoplus 11)) \bigoplus 11$$

其中：**a**为每位原数字分别乘加权因子后的结果，**m**最终为校验码，**n**为每位数字的位数。

用这个算法算出来的校验码，取值范围为0~10，其中10使用罗马数字“**X**”来表示。

基于本算法制作的身份号码校验码推算程序（源代码）[5]：

```
#include <bits/stdc++.h>
using namespace std;
int main(){
    int a[20] = {7,9,10,5,8,4,2,1,6,3,7,9,10,5,8,4,2};
    int b[20],c;
    for(int i = 0; i < 17; i++){
        cin >> b[i];
    }
    for(int i = 0; i < 17; i++){
        c += a[i] * b[i];
    }
    cout << (12-(c%11))%11;
    return 0;
}
```

```
import os
a = [7,9,10,5,8,4,2,1,6,3,7,9,10,5,8,4,2]
b = list(input())
c = 0
for i in range(0,17):
    c += a[i] * int(b[i])
def mod(x):
    return (12-(c%11))%11
print(mod(c))
os.system("pause")
```

```
indo:{
    int list a[7,9,10,5,8,4,2,1,6,3,7,9,10,5,8,4,2]
    int b,c
    l = input(b.list)
    range(0,17):{
        c += a[RANGE_I] * b[RANGE_I]
    }
    public void mod(x):{
        return (12-(c%11))%11
    }
    output(mod(c))
}
```

---

1. 本文编写依据：ISO 7064:1983 ↩
2. 由于编辑器限制，本文全文以⊕符号表取余，即取两数相除的余数 ↩ ↩ ↩ ↩ ↩
3. 信息来源：全国标准信息公共服务平台 ↩
4. 数据来源：自己算的(↩
5. 程序代码由本人临时编写，所以代码尚未规范 ↩