小秘密

【问题描述】

小强和阿米巴是好朋友。他们从小一起长大，一起度过了无忧无虑的童年时光。他们小时候非常喜欢一起玩游戏。比如，猜硬币。他们在一起抛硬币，之后猜硬币是正面落地还是反面落地。小强在猜硬币的过程中逐渐占了上风，猜的准确率总能比阿米巴高一点。

阿米巴为了弄清楚小强为什么总能赢，把他们所有的掷币结果都记录在一个文件中了。这个文件是一个长长的 01 二进制串，1 表示正面，0 表示反面。为了不让小强发现他的行为，阿米巴把这个文件加密了。阿米巴用的是他自己设计的一套算法：

这个算法有一个参数N，两个密钥K_1和K_2。它还需要一个寄存器X。

其算法如下：

1	$X_0 = K_1$
2	依次读入输入的每一个位I_p，并对它进行以下计算
3	$T_p = w\left(f\left(X_p\right)\text{and } K_2\right)$
4	$O_p = I_p \text{ xor } T_p$
5	$X_{p+1} = \left(2X_p + O_p\right)\text{mod } 2^N$
6	输出O_p

其中，

$$f(x) = (1994x^5 + 11x^4 + 4x^3 + 1995x^2 + 9x + 24)\text{mod } 2^N$$

and表示按位与运算（C/C++中的&运算），xor表示异或运算（C/C++中的^运算），mod 表示取模运算（C/C++中的%运算）。

$w(x)$表示x的二进制位中1的个数的奇偶性。若x含有奇数个1，则$w(x) = 1$；若x含有偶数个1，则$w(x) = 0$。比如，$w(13) = w(1101_2) = 1$，$w(12) = w(1100_2) = 0$。

阿米巴发现这个算法的加密效果不错。

现在，小强和阿米巴都长大了。他们进入了不同的大学，从此便很久才能见到对方。久而久之，阿米巴忘记了当初加密这个文件的密钥，只记得K_1和K_2代表了他们之间的一个小秘密。他知道N不会很大，而K_1和K_2都是不超过2^N的 64 位二进制数。他很想通过密文恢复出这组密钥。但是，学过代数学以及概率论的他知道，这几乎是不可能的。于是，他找到了小强。

小强听后露出了诡秘的笑容。

“你知道为什么我能猜的比你准吗？”

“为什么？”

“因为这个硬币不是均匀的；它出现反面的概率是$\frac{1}{2}+b$，比正面高。”

阿米巴立刻看到了恢复出密钥的希望。

“不过，给我足够长的密文，我只能恢复出K_2”

“嗯，这也很好了。”

【输入格式】

输入文件 ***crypto.in***。它的长度是 1048576 字节，表示一个 8388608 位的二进制数。高位在前，低位在后。比如 01100001011011010110010 记录下来就成了“amb”三个字节。这个文件表示加密之后的文件内容。

【输出格式】

输出文件 ***crypto.out***。它是一行 64 个 0 或 1，表示K_2的二进制表示。

【样例输入输出】

因为输入输出文件比较大，你会得到一个数据生成器，其路径为 ***crypto/cryptogen.py***。使用它的方法是，首先进入终端，在终端中运行下面的命令进入本题的文件夹：

```
cd crypto
```

然后运行如下的命令

```
python cryptogen.py N b
```

其中，N和b是算法的参数，比如

```
python cryptogen.py 50 0.3
```

耐心等待一段时间，你就能够得到输入文件 ***crypto.in***，标准输出文件 ***crypto.ans***。明文以及其他信息会保存在 ***crypto.log*** 中以方便你的调试。

注意数据生成器具有随机性，***输入相同的参数几乎总是会得到不同的结果***，请注意保管好每次生成的样例数据。

【如何读入二进制文件】

在 C/C++中，可以使用函数 fread 来读/写二进制文件。具体的使用方法可以在终端中运行以下命令来查看

```
man fread
```

下面是一个实例：

```
#include <stdio.h>
int main()
{
    FILE *fin=fopen("crypto.in","rb");
    static unsigned char data[1<<20];
    fread(data,1,1<<20,fin);
    return 0;
}
```

对于 Pascal 选手，下面这个实例也许对你有帮助。

```
program crypto;
var
  MyFile: file;
  Data: array [0..999999] of Byte;
begin
  Assign(MyFile, 'crypto.in');
  Reset(MyFile, 1);
  BlockRead(MyFile, Data, SizeOf(Data));
  Close(MyFile);
end.
```

【数据规模和约定】

生成评测所用的 10 个测试点的参数如下

测试点编号	N	b
1	5	0.12
2	20	0.12
3	50	0.5
4	50	0.4
5	50	0.35
6	50	0.3
7	60	0.2
8	60	0.15
9	60	0.13
10	60	0.12

【提示】

如果四个事件各自以0.62的概率发生，那么恰好奇数个事件的概率大约是0.49834112。为了以很高的把握区分出一个硬币出现正面的概率到底是$\frac{1}{2}$还是$\frac{1}{2}-d$，需要抛的硬币的次数正比于d^{-2}。