

SEAPROAR: Sereja and Random Array

题目描述

佳佳需要生成伪随机二进制序列。佳佳现在有两种方法，一种是线性同余方法（**linear congruential generators (LCGs)**），另一种亦或位移位移（**Xorshift**）。

佳佳现在有一些之前生成的随机二进制序列，他想知道这些序列分别是通过哪些方法生成的。你可以帮助他吗？

假设序列的长度为 N ，随机数种子是 S ，生成的序列式 $A[1], A[2], \dots, A[N]$ 。

第一个生成器如下：

```
/* ----- start here -----*/
unsigned X; // we assume that unsigned is a 32bit integer type

void srand1(unsigned S){
    X = S;
}

unsigned nextInteger1(void){
    X = X * 1103515245 + 12345;
    return (X / 65536) % 32768;
}

void generator1(int N, unsigned S, int A[]){
    srand1(S);
    for(int i=1; i<=N; i++){
        A[i] = nextInteger1() % 2;
    }
}
/* ----- end here -----*/
```

第二个生成器如下：

```
/* ----- start here -----*/
unsigned x, y, z, w; // we assume that unsigned is a 32bit integer type

void srand2(unsigned S){
    x = S;
    y = x * S;
    z = y * S;
```

```

    w = z * S;
}

unsigned nextInteger2(void) {
    unsigned t = x ^ (x << 11);
    x = y; y = z; z = w;
    return w = (w ^ (w >> 19)) ^ (t ^ (t >> 8));
}

void generator2(int N, unsigned S, int A[]) {
    srand2(S);
    for(int i=1; i<=N; i++) {
        A[i] = nextInteger2() % 2;
    }
}
/* ----- end here ----- */

```

注意，LCG 所使用的第一个生成器和 [ISO/IEC 9899 \(pp. 346–347\)](#) 所推荐的一样。Xorshift 所使用的第二个生成器和 [Marsaglia \(July 2003\)](#) 的这篇论文中所使用的一样。

输入格式

输入数据的第一行包含一个整数 T ，表示数据组数。对于每组测试数据，仅包含一行，包含一个长度为 N 的 01-字符串。

输出格式

对于每组数据，输出一行表示对应的答案。如果是第一种则输出“LCG”，如果是第二种输出“Xorshift”。

数据范围

- $1 \leq T \leq 30$
- 不存在一组种子 (s, t) ，满足 $0 \leq s, t \leq 10^9$ ， $\text{generator1}(N, s, A)$ 和 $\text{generator2}(N, t, A)$ 所产生的结果相同，因而答案是唯一的。

子任务一：10分

- $50 \leq N \leq 500$ 。
- 可以在 $0 \leq s \leq 500$ 的范围内找到解。

子任务二: 40 分

- $500 \leq N \leq 100000$ 。
- 可以在 $0 \leq s \leq 31313$ 的范围内找到解。

子任务三: 20 分

- $100000 \leq N \leq 200000$ 。
- 可以在 $0 \leq s \leq 10^9$ 的范围内找到解。

子任务四: 30 分

- $500 \leq N \leq 200000$ 。
- 可以在 $0 \leq s \leq 10^9$ 的范围内找到解。

样例数据

输入

```
6
1101100100101111010011010101110100001000000101001110101011010101010
000101101110101101110110010111000000011001101110101
11010100010110001101010110111000010001110010010011011110010010110000001100110
01011010100111100111101001010010100100111000111110
0000000000000000000000001001001010101011001111101101010
1110100101000000011110100111111000010000111010011111000001111
```

输出

```
LCG
LCG
LCG
Xorshift
Xorshift
Xorshift
```

样例解释

Example 1. `generator1(67, 5, A)` 生成了这个序列。
 Example 2. `generator1(51, 8, A)` 生成了这个序列。
 Example 3. `generator1(77, 58, A)` 生成了这个序列。

Example 4. `generator2(50, 5, A)` 生成了这个序列。

Example 5. `generator2(55, 8, A)` 生成了这个序列。

Example 6. `generator2(62, 58, A)` 生成了这个序列。

Problem Setter: Sergey Nagin

Problem Tester: Hiroto Sekido

Translated by : Minako Kojima