Sereja and Random Array

Problem code: SEAPROAR

Read problems statements in Mandarin Chinese and Russian.

Sereja likes to generate pseudo random binary sequences. Now Sereja has two generators: one is a based on linear congruential generators (LCGs) and another is based on Xorshift.

Sereja has some binary sequences generated in past times, and he wants to know which generator makes these sequences. You can know the details of Sereja's generators, then can you answer this problem?

The following is the details. We may give the length **N** and a seed integer **S** to the generators, then they generate a binary sequence **A[1], A[2], ..., A[N]**.

The 1st generator works as follows (C++ code. If you are not familiar with C++, please see the below section **Notes for C++**):

```
/* ------------------ 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 ----------------------*/
```

The 2nd generator works as follows (C++ code):

```
/* ------------------ 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 ----------------------*/
```

Note that the LCG used in the 1st generator is the same one suggested in ISO/IEC 9899 (pp. 346--347), and Xorshift used in the 2nd generator is the same one in the paper by Marsaglia (July 2003).

Input

The first line of input contains an integer **T**, denoting the number of test cases. Then **T** test cases follow.

Each test case has only one line. The line contains the string of length **N**, denoting the array **A[1], A[2], ..., A[N]**, where the string consists of only characters '**0**' and '**1**', and the i^{th} character denotes **A[i]**.

Note that the integer **N** is not given in the input explicitly.

Output

SUCCESSFUL SUBMISSIONS

User	Score	Mem	Lang	Solution
avmnusng	100.000	2.1M	C	View
gvaibhav21	100.000	2.1M	C	View
mediocoder	100.000	2.1M	C	View
npriyadarshi	100.000	2.2M	C	View
additya1998	100.000	2.8M	C++ 4.3.2	View
mgch	100.000	2.8M	C++ 4.3.2	View
ishraq	100.000	2.9M	C++ 4.3.2	View
rajat1603	100.000	2.9M	C++ 4.3.2	View
manish05	100.000	3M	C++ 4.3.2	View
shashank_kgp	100.000	3.1M	C++ 4.9.2	View
ashish1610	100.000	3.2M	C++ 4.9.2	View
xorfire	100.000	3.2M	C++14	View

1 of 6 Next »

For each test case, print "LCG" if the given sequence generated by the 1st generator, or print "Xorshift" if the given sequence is generated by the 2nd generator.

Constraints and Subtasks

- **1 ≤ T ≤ 30**
- There is no pair of integers **(s, t)** such that **0 ≤ s, t ≤ 10^9** and both **generator1(N, s, A)** and **generator2(N, t, A)** generate the given sequence. (Thus the answer will be determined uniquely)

Subtask 1 (10 points)

- **50 ≤ N ≤ 500**
- There is an integer **0 ≤ s ≤ 500** such that **generator1(N, s, A)** or **generator2(N, s, A)** generates the given sequence.

Subtask 2 (40 points)

- **500 ≤ N ≤ 100000**
- There is an integer **0 ≤ s ≤ 31313** such that **generator1(N, s, A)** or **generator2(N, s, A)** generates the given sequence.

Subtask 3 (20 points)

- **100000 ≤ N ≤ 200000**
- There is an integer **0 ≤ s ≤ 10^9** such that **generator1(N, s, A)** or **generator2(N, s, A)** generates the given sequence.

Subtask 4 (30 points)

- **500 ≤ N ≤ 200000**
- There is an integer **0 ≤ s ≤ 10^9** such that **generator1(N, s, A)** or **generator2(N, s, A)** generates the given sequence.

Example

Input:
6
11011001001011110100110101011101000100000010100111010101101010101010
00010110111010110111011001011100000001100110111010 1
11010100010110001101010110111000010001110010010011011110010010110000001100110
01011010100111100111101001010010100100111000111110
00000000000000000000000001001001010101011001111101101010
11101001010000000111101001111111000010000111010011111000001111

Output:
LCG
LCG
LCG
Xorshift
Xorshift
Xorshift

Explanation

Example 1. generator1(67, 5, A) generates the given sequence.

Example 2. generator1(51, 8, A) generates the given sequence.

Example 3. generator1(77, 58, A) generates the given sequence.

Example 4. generator2(50, 5, A) generates the given sequence.

Example 5. generator2(55, 8, A) generates the given sequence.

Example 6. generator2(62, 58, A) generates the given sequence.

Notes for C++

At first, in the codes, almost every operation will be done with unsigned.

Thus operations will return the result modulo 2^{32}.
For example,

```
X * 1103515245 + 12345
```

means that
(X × 1103515245 + 12345) mod 2^{32},

and

```
(X / 65536) % 32768
```

means that
(floor(X / 65536) mod 32768) mod 2^{32},

in terms of mathematical notations.

Then there are some bit operations in the 2nd generator.
The operators **<<** and **>>** denote bit shifts.
For example,

```
X << 15
```

means that
(X × 2^{15}) mod 2^{32},

and

X >> 13

means that
floor(X / 2^{13}).

And the operator **^** denotes bitwise XOR.

Author:	sereja
Tester:	laycurse
Date Added:	29-11-2014
Time Limit:	1 sec
Source Limit:	50000 Bytes
Languages:	ADA, ASM, BASH, BF, C, C99 strict, CAML, CLOJ, CLPS, CPP 4.3.2, CPP 4.9.2, CPP14, CS2, D, ERL, FORT, FS, GO, HASK, ICK, ICON, JAVA, JS, LISP clisp, LISP sbcl, LUA, NEM, NICE, NODEJS, PAS fpc, PAS gpc, PERL, PERL6, PHP, PIKE, PRLG, PYTH, PYTH 3.1.2, RUBY, SCALA, SCM guile, SCM qobi, ST, TCL, TEXT, WSPC

SUBMIT

Comments

cyberax *@ 6 Mar 2015 03:58 PM*

@admin: why lines are 0-prefixed ?

dpraveen *@ 6 Mar 2015 04:16 PM*

@cyberax: Can you please explain your query in more detail?

cyberax *@ 6 Mar 2015 05:02 PM*

@admin: Input section "The line contains the string of length N, denoting the array A[1], A[2], ..., A[N],". But there is a '0' before (probably related to A[0]). Thus It's A[0] to A[N-1], not A[1] to A[N].

dpraveen *@ 6 Mar 2015 05:23 PM*

@cyberax: No, everything is as it is stated in the problem statement.

laycurse *@ 6 Mar 2015 08:32 PM*

Sorry, it seems my mistakes. I'll fix it as soon as possible.

laycurse *@ 6 Mar 2015 08:59 PM*

I have fixed the judge test cases. The statements will be fixed soon. And the submissions will be rejudged. The sample input should be 6
1101100100101111010011010101110100001000000101001110101011010101010
000101101110101101110110010111000000011001101110101
11010100010110001101010110111000010001110010010011011110010010110000001100110
0101101010011110011110100101001010010011100011110
000000000000000000000000100100101010101100111101101010
1110100101000000011110100111111100001000011101001111000001111

ashish1610 *@ 8 Mar 2015 01:56 AM*

Nice problem :)

Need help? Post a comment. But before that please spare a moment to read the guidelines.

Your name:
shubhmsng

Comment: *

Save