

SEAPROAR: Sereja and Random Array

Легенда, описанная далее, переформулирована и упрощена переводчиком, чтобы читатель мог лучше понять условие задачи. Оригинальную легенду вы можете прочитать на странице задачи в контексте.

Условие:

Сережа любит генерировать псевдослучайные бинарные последовательности. Сейчас у Сережи есть два генератора: первый из них основан на линейном конгруэнтном методе ([LCGs](#)), а второй известен как [Xorshift](#).

У Сережи есть сгенерированная бинарная последовательность, и он хочет узнать, каким методом она была сгенерирована. Помогите Сереже сделать это.

Далее некоторые пояснения к методам генерации последовательностей.

В генератор может быть передана длина N и начальное значение целое число S . Исходя из этих данных, будет сгенерирована бинарная последовательность A_1, A_2, \dots, A_N .

Первый генератор работает следующим образом (код написан на языке 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 -----*/
```

Второй генератор работает следующим образом (код написан на языке C++):

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

Отметим, что используемый в первом генераторе метод **LCG** аналогичен предложенному в ISO/IEC 9899 (стр. 346--347), и метод **Xorshift**, используемый во втором генераторе, аналогичен методу в публикации Marsaglia (июнь 2003).

Формат ввода:

В первой строке содержится целое число **T** – количество тестов.

Далее следует описание тестов в следующем формате:

В первой (и единственной) строке содержится сгенерированная последовательность из **N** чисел A_1, \dots, A_N , каждое A_i равно либо нулю, либо единице.

Отметим, что длина последовательности **N** во входных данных явно не задается.

Формат вывода:

Для каждого тестового случая выведите в отдельную строку “**LCG**”, если данная последовательность сгенерирована первым генератором, или “**Xorshift**”, если она сгенерирована вторым.

Оценивание & ограничения:

- $1 \leq T \leq 30$
- Не существует пары целых чисел (s, t) таких, что $0 \leq s, t \leq 10^9$ и оба метода **generator1(N, s, A)** и **generator2(N, t, A)** генерируют заданную последовательность (то есть ответ определен однозначно).

Подзадача 1 (10 баллов)

- $50 \leq N \leq 500$
- Существует целое число $0 \leq s \leq 500$ такое, что **generator1(N, s, A)** или **generator2(N, t, A)** генерирует заданную последовательность.

Подзадача 2 (40 баллов)

- $500 \leq N \leq 10^5$
- Существует целое число $0 \leq s \leq 31313$ такое, что **generator1(N, s, A)** или **generator2(N, t, A)** генерирует заданную последовательность.

Подзадача 3 (20 баллов)

- $10^5 \leq N \leq 2 * 10^5$
- Существует целое число $0 \leq s \leq 10^9$ такое, что **generator1(N, s, A)** или **generator2(N, t, A)** генерирует заданную последовательность.

Подзадача 4 (30 баллов)

- $500 \leq N \leq 2 * 10^5$
- Существует целое число $0 \leq s \leq 10^9$ такое, что **generator1(N, s, A)** или **generator2(N, t, A)** генерирует заданную последовательность.

Примеры тестов:

Входные данные:

