



### Nice3. Симпатичные узоры — 3

Имя входного файла: nice3.in  
 Имя выходного файла: nice3.out

Компания *Broken Tiles* планирует заняться выкладыванием во дворах у состоятельных клиентов узоров из черных и белых плиток, каждая из которых имеет размер  $1 \times 1$  метр. Известно, что дворы всех состоятельных людей имеют наиболее модную на сегодня форму прямоугольника  $n \times m$  метров.

Однако при составлении финансового плана у директора этой организации появилось целых две серьезные проблемы: во-первых, каждый новый клиент, естественно, захочет, чтобы узор, выложенный у него во дворе, отличался от узоров всех остальных клиентов этой фирмы, а во-вторых, этот узор должен быть симпатичным.

Как показало исследование, узор является симпатичным, если в нем нигде не встречается квадрата  $2 \times 2$  метра, полностью покрытого плитками одного цвета.

Для составления финансового плана директору необходимо узнать, сколько клиентов он сможет обслужить, прежде чем симпатичные узоры данного размера закончатся. Помогите ему!

#### Формат входного файла

На первой строке входного файла находятся три натуральных числа  $n$  и  $m$  ( $1 \leq n \leq 10^{100}$ ,  $1 \leq m \leq 5$ ,  $1 \leq p \leq 10\,000$ ).

#### Формат выходного файла

Выведите в выходной файл единственное число — количество различных симпатичных узоров, которые можно выложить во дворе размера  $n \times m$ , взятое по модулю  $p$ . Узоры, получающиеся друг из друга сдвигом, поворотом или отражением, считаются различными.

#### Пример

nice3.in	nice3.out
2 2 20	14
3 3 7	0

### RNG. Генератор псевдослучайных чисел

Имя входного файла: rng.in  
 Имя выходного файла: rng.out

Последовательность псевдослучайных чисел  $X_1, X_2, \dots, X_i, \dots$  генерируется следующим образом: числа  $X_1, X_2, \dots, X_k$  задаются в явном виде, а каждое следующее вычисляется по формуле:  $X_n = (a_1 X_{n-1} + a_2 X_{n-2} + \dots + a_k X_{n-k} + b) \bmod m$ . Вы должны написать программу, вычисляющую  $N$ -е число этой последовательности.

#### Формат входного файла

Во входном файле записаны целые числа в следующем порядке:  $k$  ( $1 \leq k \leq 30$ ),  $m$  ( $1 \leq m \leq 1000$ ),  $a_1, \dots, a_k$  ( $0 \leq a_i < m$ ),  $b$  ( $0 \leq b < m$ ),  $X_1, \dots, X_k$  ( $0 \leq X_i < m$ ),  $N$  ( $1 \leq N \leq 10^{100}$ ). Числа разделяются пробелами и (или) символами перевода строки.

### Формат выходного файла

В выходной файл нужно вывести одно число —  $X_N$ .

#### Пример

rng.in	rng.out
2 5 1 2 3 4 0 3	1

### Gauss. Система линейных уравнений

Имя входного файла: gauss.in  
 Имя выходного файла: gauss.out

Жюри потребовалось решить систему линейных уравнений:

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = b_1, \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n = b_2, \\ \dots \\ a_{n1}x_1 + a_{n2}x_2 + \dots + a_{nn}x_n = b_n. \end{cases}$$

Помогите им в этом — напишите программу, решающую эту систему.

#### Формат входного файла

Первая строка входного файла содержит одно число  $n$  ( $1 \leq n \leq 100$ ). Каждая из следующих  $n$  строк содержит описание очередного уравнения —  $n + 1$  чисел  $a_{i1}, a_{i2}, \dots, a_{in}, b_i$ . Все эти числа вещественные. Гарантируется, что система имеет ровно одно решение.

#### Формат выходного файла

В выходной файл выведите числа  $x_1, x_2, \dots, x_n$  как можно точнее.

#### Пример

gauss.in	gauss.out
2 1 1 3 1 -1.5 0.5	2.0000 0.9999999999999999