Задача A. Плохая подстрока

Имя входного файла:	`badsubs.in`
Имя выходного файла:	`badsubs.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мебибайт

Найдите, сколько существует строк заданной длины n, состоящих только из символов «a», «b» и «c» и не содержащих подстроки «ab».

Формат входных данных

Во входном файле задано n ($0 \leqslant n \leqslant 22$).

Формат выходных данных

Выведите количество таких строк.

Примеры

badsubs.in	**badsubs.out**
0	1
3	21
11	46368

Задача B. Число Фибоначчи

Имя входного файла:	`fiblong.in`
Имя выходного файла:	`fiblong.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мебибайт

Числа Фибоначчи $F_0, F_1, F_2, \ldots, F_n$ определяются следующим образом: $F_0 = 0$, $F_1 = 1$, а для любого $n > 1$ выполнено равенство $F_n = F_{n-1} + F_{n-2}$.

По заданному числу n выведите число Фибоначчи F_n.

Формат входных данных

В первой строке входного файла задано единственное число n ($0 \leqslant n \leqslant 92$).

Формат выходных данных

Выведите число F_n в первой строке выходного файла.

Примеры

fiblong.in	**fiblong.out**
1	1
2	1
3	2
5	5

Задача C. Числа Фибоначчи по модулю

Имя входного файла:	`fibmod.in`
Имя выходного файла:	`fibmod.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мебибайт

Числа Фибоначчи $F_0, F_1, F_2, \ldots, F_n$ определяются следующим образом: $F_0 = 0$, $F_1 = 1$, а для любого $n > 1$ выполнено равенство $F_n = F_{n-1} + F_{n-2}$.

По заданному числу n выведите остаток от деления числа Фибоначчи F_n на m.

Формат входных данных

В первой строке ввода заданы через пробел два целых числа n и m ($0 \leqslant n \leqslant 10^9$, $1 \leqslant m \leqslant 10^9$).

Формат выходных данных

В первой строке выведите одно число: остаток от деления F_n на m.

Пример

fibmod.in	**fibmod.out**
8 12345	21
100 1000000000	261915075

Пояснение к примеру

Первые несколько чисел Фибоначчи таковы: $F_0 = 0$, $F_1 = 1$, $F_2 = 1$, $F_3 = 2$, $F_4 = 3$, $F_5 = 5$, $F_6 = 8$, $F_7 = 13$, $F_8 = 21$.

Число Фибоначчи с номером 100 равно 354 224 848 179 261 915 075. Остаток от деления этого числа на 1 000 000 000 — это последние девять цифр его десятичной записи.

Задача D. Числа Фибоначчи по модулю-2

Имя входного файла:	`fibmod2.in`
Имя выходного файла:	`fibmod2.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мебибайт

Числа Фибоначчи $F_0, F_1, F_2, \ldots, F_n$ определяются следующим образом: $F_0 = 0$, $F_1 = 1$, а для любого $n > 1$ выполнено равенство $F_n = F_{n-1} + F_{n-2}$.

По заданному числу n выведите остаток от деления числа Фибоначчи F_n на m.

Формат входных данных

В первой строке ввода заданы через пробел два целых числа n и m ($0 \leqslant n \leqslant 10^{18}$, $1 \leqslant m \leqslant 10^{18}$).

Формат выходных данных

В первой строке выведите одно число: остаток от деления F_n на m.

Пример

fibmod2.in	**fibmod2.out**
8 12345	21
100 1000000000	261915075

Пояснение к примеру

Первые несколько чисел Фибоначчи таковы: $F_0 = 0$, $F_1 = 1$, $F_2 = 1$, $F_3 = 2$, $F_4 = 3$, $F_5 = 5$, $F_6 = 8$, $F_7 = 13$, $F_8 = 21$.

Число Фибоначчи с номером 100 равно 354 224 848 179 261 915 075. Остаток от деления этого числа на 1 000 000 000 — это последние девять цифр его десятичной записи.

Задача E. Тест Миллера—Рабина

Имя входного файла:	`millerrabin.in`
Имя выходного файла:	`millerrabin.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мебибайт

Существует множество способов проверки числа на простоту. Например, если проверяемое число N достаточно мало, то можно просто поделить N на все простые числа, не превосходящие $\sqrt{N}$. Если N делится нацело хотя бы на одно из них — значит, оно составное, в противном же случае оно является простым.

Однако, когда число N велико, такой метод может потребовать от проверяющего слишком много времени — ведь трудоёмкость растёт экспоненциально от длины числа N. В настоящее время известно несколько способов определить простоту числа точно, но все они, как правило, работают слишком долго.

Поэтому чаще применяют способы, определяющие простоту числа с некоторой вероятностью. Один из наиболее быстрых и вместе с тем довольно надёжных способов известен как тест Миллера—Рабина. Ознакомимся с ним подробнее.

Сначала проверим, что N нечётно и больше, чем 1 (в противном случае проверка тривиальна). Представим $N - 1$ как $2^s \cdot d$; заметим, что $s \geqslant 1$.

Теперь для нескольких различных $a \in [1, N-1]$ произведём следующую процедуру. Рассмотрим числа $k_r = a^{2^r \cdot d}$ для $r = 0, 1, \ldots, s-1$. Если $k_0 \mod N \neq 1$ и ни одно из k_r не совпадает с -1 по модулю N (другими словами, $k_r \mod N \neq N-1$), число N — составное. В противном случае мы повторяем эту процедуру для следующего a. Чем больше чисел a было проверено, тем больше вероятность того, что число N — простое. Обычно в качестве a подставляют первые несколько простых чисел — $2, 3, 5, 7, 11, \ldots$.

Мы не будем сейчас останавливаться на том, почему тест Миллера—Рабина работает. Наша задача заключается в другом — по числу N определить, каково же наименьшее простое число a, для которого описанная выше процедура приведёт к установлению того, что N — составное (разумеется, если это так). Число a не окажется слишком большим — известно, что наименьшее нечётное составное число N такое, что для него не срабатывают проверки с $a = 2, 3, 5, 7, 11, 13, 17, 19$, равно $341\,550\,071\,728\,321$.

Формат входных данных

В первой строке входного файла записано число N ($1 \leqslant N \leqslant 10^{14}$).

Формат выходных данных

Если N чётно или равно единице, и тест Миллера—Рабина неприменим, выведите -1. Если число N нечётное и простое, выведите 0. Иначе выведите наименьшее простое число a такое, что при его проверке по приведённому выше алгоритму выяснится, что N — составное.

Примеры

millerrabin.in	millerrabin.out
2	-1
15	2
4	-1
821	0
2047	3

Задача F. Первообразный корень

Имя входного файла:	primitive.in
Имя выходного файла:	primitive.out
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мебибайт

Мультипликативный порядок числа q по модулю m — это минимальное натуральное число k такое, что $q^k \mod m = 1$.

Число q называется *первообразным корнем* по простому модулю p, если мультипликативный порядок q равен $p-1$.

Дано простое число p и набор чисел $q_1, q_2, \ldots, q_n$. Для каждого q_i выясните, является ли оно первообразным корнем по модулю p.

Формат входных данных

В первой строке входного файла заданы через пробел два целых числа n и p — количество чисел и модуль ($1 \leqslant n \leqslant 100$, $2 \leqslant p \leqslant 10^9$, число p является простым). Следующие n строк содержат по одному числу q_i каждая ($0 < q_i < p$).

Формат выходных данных

Выведите в выходной файл n строк; в i-й строке выведите «YES», если q_i является первообразным корнем по модулю p, и «NO» в противном случае.

Примеры

primitive.in	primitive.out
1 3 2	YES
2 7 2 3	NO YES

Задача G. Произведение матриц

Имя входного файла:	product.in
Имя выходного файла:	product.out
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мебибайт

Произведением матриц A и B размера $p \times q$ и $q \times r$, соответственно, называется матрица C размера $p \times r$, элементы которой вычисляются по формуле

$$C_{i,j} = \sum_{k=1}^{q} A_{i,k} \cdot B_{k,j}.$$

По данным матрицам A и B найдите их произведение.

Формат входных данных

В первой строке входного файла заданы через пробел три целых числа p, q и r ($1 \leqslant p, q, r \leqslant 100$). В следующих p строках записана матрица A; каждая из этих строк содержит q целых чисел, разделённых пробелами. Наконец, в последних q строках записана матрица B; каждая из этих строк содержит r целых чисел, разделённых пробелами. Элементы матриц не превосходят 100 по абсолютной величине.

Формат выходных данных

В выходной файл выведите матрицу C — p строк, в каждой из которых — r чисел через пробел.

Примеры

product.in	product.out
2 2 2 1 0 0 1 1 0 0 1	1 0 0 1
1 3 1 1 2 3 -1 -2 -3	-14
3 2 4 0 1 1 0 0 1 2 1 0 0 1 1 2 1	1 1 2 1 2 1 0 0 1 1 2 1