

Задача А. Волшебные ночи

Имя входного файла: magic.in
Имя выходного файла: magic.out
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

В этой задаче нужно найти количество волшебных ночей в году на далёкой планете.

Вокруг далёкой планеты Этан вращается три луны: Клементина, Лея и Матильда. Каждую k -ю ночь наступает полнолуние Клементины, каждую l -ю ночь — полнолуние Леи, а каждую m -ю ночь — полнолуние Матильды. В году на этой планете n ночей, а Новый Год наступает днём.

Ночь на планете Этан считается волшебной, если в эту ночь наступает полнолуние хотя бы у одной из лун. Известно, что в последнюю ночь прошлого года полнолуние наступило одновременно у всех трёх лун Этана. Сколько волшебных ночей в текущем году?

Формат входных данных

В первой строке заданы четыре целых числа k, l, m и n ($1 \leq k, l, m, n \leq 10^9$). Числа разделены пробелами.

Формат выходных данных

В первой строке выведите одно целое число — количество волшебных ночей в текущем году.

Примеры

magic.in	magic.out
3 4 5 10	7
5 5 5 10	2
30 29 31 360	35
2 4 6 5	2

Пояснения к примерам

В первом примере волшебными считаются 3-я, 4-я, 5-я, 6-я, 8-я, 9-я и 10-я ночи.

Во втором примере волшебных ночей только две — 5-я и 10-я ночи.

В третьем примере волшебными оказываются 12 ночей, когда полнолуние наступает у Клементины, 12 ночей, когда полнолуние наступает у Леи, и 11 ночей, когда полнолуние наступает у Матильды.

В четвёртом примере во вторую ночь наступает полнолуние Клементины, а в четвёртую — Клементины и Леи. Поскольку в году всего пять ночей, следующее полнолуние Матильды случится только в следующем году.

Задача В. Делители

Имя входного файла: divisors.in
Имя выходного файла: divisors.out
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Натуральное число a называется *делителем* натурального числа b , если $\frac{b}{a}$ — также натуральное число. Например, 1, 2, 3 и 6 — делители числа 6, а 4, 5 и 7 не являются его делителями.

В этой задаче требуется определить, каково максимальное количество различных делителей, которое может иметь натуральное число от 1 до N , включительно, и найти минимальное из чисел на этом интервале, имеющее ровно столько делителей.

Формат входных данных

В первой строке задано число N ($1 \leq N \leq 10^{18}$).

Формат выходных данных

Выведите два целых числа через пробел — какое максимальное количество делителей может иметь натуральное число от 1 до N включительно, а также минимальное натуральное число, имеющее столько делителей.

Примеры

divisors.in	divisors.out
2	2 2
5	3 4
7	4 6
18	6 12

Задача С. Простые числа

Имя входного файла: primes.in
Имя выходного файла: primes.out
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Простым, как известно, называется натуральное число, которое делится нацело только на себя и на единицу. Число, делящееся на другое натуральное число, меньшее его, называется составным. Единица не считается ни простым, ни составным числом. Так, есть 25 простых чисел, не превосходящих 100 — это числа 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97.

В этой задаче мы попробуем выяснить, сколько простых чисел расположено на отрезке $[A, B]$, где A и B — целые и $A \leq B$. Математики интересовались подобными вопросами уже давно. Ещё в середине XIX века француз Джозеф Луи Франсуа Бертран выдвинул гипотезу о том, что для любого $n > 1$ между n и $2n$ есть по крайней мере одно простое число. Эта гипотеза была впоследствии доказана Пафнутием Львовичем

Чебышёвым и получила название теоремы Чебышёва. Другая теорема, связывающая имена этих двух математиков, говорит о том, что количество простых чисел от 1 до n ведёт себя примерно как $\frac{n}{\ln n}$.

Возможности современных вычислительных машин позволяют посчитать количество простых чисел от A до B точно, если A и B достаточно невелики. В этом и состоит предлагаемая задача.

Формат входных данных

В первой строке записаны два числа — A и B ($1 \leq A \leq B \leq 50\,000\,000$).

Формат выходных данных

Выведите одно число — количество простых чисел на отрезке $[A, B]$.

Примеры

<code>primes.in</code>	<code>primes.out</code>
1 2	1
2 3	2
1 100	25
98 98	0
97 97	1

Задача D. Тест Миллера–Рабина

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

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 $s \geq 1$.

Теперь для нескольких различных $a \in [1, N - 1]$ произведём следующую процедуру. Рассмотрим числа $k_r = a^{2^r \cdot d}$ для $r = 0, 1, \dots, s - 1$. Если $k_0 \bmod N \neq 1$ и ни одно из k_r не совпадает с -1 по модулю N (другими словами, $k_r \bmod N \neq N - 1$), 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, ...

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 ($1 \leq N \leq 10^{14}$).

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

<code>millerrabin.in</code>	<code>millerrabin.out</code>
2	-1
15	2
4	-1
821	0
2047	3

Задача E. Обратный элемент

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

Дано целое число $n > 0$. Рассмотрим множество Z_n , элементами которого являются целые числа $0, 1, 2, \dots, n - 1$. Элемент q называется *обратным* к элементу p , если $(p \cdot q) \bmod n = 1$.

Найдите обратный элемент q по заданным числам n и p .

Формат входных данных

В первой строке входного файла заданы два целых числа n и p через пробел ($1 \leq n \leq 10^9$, $0 \leq q < n$).

Формат выходных данных

Если у числа p нет в множестве Z_n обратного элемента, выведите -1 . В противном случае выведите q .

Примеры

inverse.in	inverse.out
3 2	2
5 2	3
8 4	-1
17 0	-1

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

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 ($1 \leq p, q, r \leq 100$). 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.

Задача F. Первообразный корень

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

Мультипликативный порядок числа q по модулю m — это минимальное целое положительное число k , для которого $q^k \bmod m = 1$.

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, \dots, q_n . Для каждого q_i выясните, является ли оно первообразным корнем по модулю p .

Формат входных данных

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

Формат выходных данных

Выведите 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 мегабайт

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

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

Задача H. Числа Фибоначчи по модулю-2

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

Числа Фибоначчи $F_0, F_1, F_2, \dots, F_n$ 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 ($0 \leq n \leq 10^{18}$, $1 \leq m \leq 10^{18}$).

Формат выходных данных

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

Примеры

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

Пояснения к примерам

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 I. Зеркальный лабиринт

Имя входного файла:	mirror-labyrinth.in
Имя выходного файла:	mirror-labyrinth.out
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

В зеркальном лабиринте живут солнечные зверьки: зайчики, кролики и тушканчики. Пока свет в лабиринте погашен, в лабиринте нет никаких солнечных зверьков. Как только свет включают, каждую секунду происходят превращения. В k -ю секунду с момента включения одновременно происходят следующие преобразования:

- В лабиринте появляется k новых солнечных зайчиков.
- Каждый солнечный зайчик, появившийся на предыдущей секунде, исчезает, превращаясь в a кроликов.
- Каждый солнечный кролик, появившийся на предыдущей секунде, исчезает, разделяясь на несколько частей: одного зайчика, b кроликов и одного тушканчика.
- Каждый солнечный тушканчик, появившийся на предыдущей секунде, исчезает, производя на свет c зайчиков и трёх кроликов.

Кроме того, лабиринт вмещает ограниченное количество зверьков каждого типа. Поэтому в конце каждой секунды — то есть после всех превращений, произошедших в течение этой секунды — если количество s зверьков какого-либо из трёх типов больше или равно m , зверьков этого типа остаётся $s \bmod m$ (остаток от целочисленного деления s на m).

По заданным числам n , m , a , b и c найдите, сколько зверьков каждого из трёх типов в отдельности окажется в лабиринте после того, как свет будет включён в течение n секунд.

Формат входных данных

В первой строке ввода заданы через пробел пять целых чисел: n , m , a , b и c ($1 \leq n \leq 10^9$, $1 \leq m \leq 10^9$, $0 \leq a, b, c < m$).

Формат выходных данных

В первой строке выведите три числа: количество солнечных зайчиков, солнечных кроликов и солнечных тушканчиков в лабиринте после того, как свет будет включён в течение n секунд. Разделяйте соседние числа в строке пробелами.

Примеры

mirror-labyrinth.in	mirror-labyrinth.out
1 10 2 3 4	1 0 0
2 10 2 3 4	2 2 0
3 10 2 3 4	5 0 2
4 10 2 3 4	2 6 0

Пояснение к примеру

В примерах представлены первые четыре секунды преобразований при $a = 2$, $b = 3$, $c = 4$ и $m = 10$.

На первой секунде появляется один солнечный зайчик.

На второй секунде он превращается в двух солнечных кроликов, а также появляются два новых солнечных зайчика.

На третьей секунде зайчики, исчезая, порождают четырёх новых кроликов, а кролики — двух зайчиков, шестерых кроликов и двух тушканчиков. Заметим, что количество солнечных кроликов становится равным $a \cdot 2 + b \cdot 2 = 2 \cdot 2 + 3 \cdot 2 = 10$. Поэтому в конце третьей секунды их численность падает до $10 \bmod 10 = 0$. Кроме того, появляется ещё три новых солнечных зайчика.

На четвёртой секунде появляется четыре новых солнечных зайчика, пять старых зайчиков, исчезая, порождают десять кроликов, а два старых тушканчика — восьмерых зайчиков и шестерых кроликов. Поскольку $m = 10$, из $4 + 8 = 12$ зайчиков и $10 + 6 = 16$ кроликов остаётся $12 \bmod 10 = 2$ зайчика и $16 \bmod 10 = 6$ кроликов.