

Задача А. Проверка на простоту

Имя входного файла: `isprime.in`
Имя выходного файла: `isprime.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

Вам доверяется важная миссия проверки чисел на простоту. Не подведите галактику!

Формат входного файла

На вход подаётся одно число — N ($2 \leq N \leq 10^9$).

Формат выходного файла

Выведите `True`, если число воистину простое, и `False`, если число составное.

Примеры

<code>isprime.in</code>	<code>isprime.out</code>
2	<code>True</code>
10	<code>False</code>

Задача В. Разложение на множители

Имя входного файла: `factor.in`
Имя выходного файла: `factor.out`
Ограничение по времени: 0.25 секунда
Ограничение по памяти: 64 мегабайта

Дано число. Требуется разложить его на простые множители.

Формат входного файла

Вводится число N ($2 \leq N \leq 10^9$).

Формат выходного файла

Выведите через пробел разложение на простые множители в порядке возрастания множителей.

Примеры

<code>factor.in</code>	<code>factor.out</code>
17	17
60	2 2 3 5

Задача С. Шестерёнки

Имя входного файла: `gears.in`
Имя выходного файла: `gears.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Даны две сцепленные шестерёнки. У одной шестерёнки N зубцов, у другой — K . Требуется найти, какое минимальное число поворотов на один зубчик требуется сделать, чтобы

шестерёнки вернулись в исходное состояние.

Формат входного файла

В единственной строке — два числа, N и K . $1 \leq N, K \leq 10^{100}$.

Формат выходного файла

Выведите искомое количество поворотов.

Примеры

<code>gears.in</code>	<code>gears.out</code>
2 3	6
6 21	42

Задача D. МегаНОД

Имя входного файла: `megagcd.in`
Имя выходного файла: `megagcd.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Дано несколько чисел. Найти самое большое число, на которое делятся все несколько чисел.

Формат входного файла

В единственной строке даны через пробел несколько чисел ($1 \leq \text{несколько} \leq 1000$, $1 \leq \text{каждое} \leq 10^9$).

Формат выходного файла

Выведите искомое число.

Примеры

<code>megagcd.in</code>	<code>megagcd.out</code>
18 30 21	3

Примечание

Ваша программа должна содержать две функции. Первая должна принимать на вход список и возвращать НОД всех чисел списка. Вторая должна вычислять НОД двух чисел.

Задача Е. Марсианские факториалы

Имя входного файла: `martian.in`
Имя выходного файла: `martian.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

В 3141 году очередная экспедиция на Марс обнаружила в одной из пещер таинственные знаки. Они однозначно доказывали существование на Марсе разумных существ. Однако смысл этих таинственных знаков долгое время оставался неизвестным. Недавно один из

ученых, профессор Очень-Умный, заметил один интересный факт: всего в надписях, составленных из этих знаков, встречается ровно K различных символов. Более того, все надписи заканчиваются на длинную последовательность одних и тех же символов.

Вывод, который сделал из своих наблюдений профессор, потряс всех ученых Земли. Он предположил, что эти надписи являются записями факториалов различных натуральных чисел в системе счисления с основанием K . А символы в конце — это конечно же нули — ведь, как известно, факториалы больших чисел заканчиваются большим количеством нулей. Например, в нашей десятичной системе счисления факториалы заканчиваются на нули, начиная с $5! = 1 \cdot 2 \cdot 3 \cdot 4 \cdot 5$. А у числа $100!$ в конце следует 24 нуля в десятичной системе счисления и 48 нулей в системе счисления с основанием 6 — так что у предположения профессора есть разумные основания!

Теперь ученым срочно нужна программа, которая по заданным числам N и K найдет количество нулей в конце записи в системе счисления с основанием K числа $N! = 1 \cdot 2 \cdot 3 \cdot \dots \cdot (N - 1) \cdot N$, чтобы они могли проверить свою гипотезу. Вам придется написать им такую программу!

Формат входного файла

В первой строке входных данных содержатся числа N и K , разделенные пробелом, ($1 \leq N \leq 10^9$, $2 \leq K \leq 1000$).

Формат выходного файла

Выведите число X — количество нулей в конце записи числа $N!$ в системе счисления с основанием K .

Примеры

	martian.in	martian.out
5 10		1
100 10		24
100 6		48
3 10		0