

Задача А. Дружественные числа

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

Два различных натуральных числа называются дружественными, если первое из них равно сумме делителей второго числа, а второе равно сумме делителей первого числа. При подсчете делителей числа учитываются все делители, кроме самого числа. Проверьте на дружественность числа N и M .

Формат входного файла

Во входном файле содержатся два натуральных числа N и M ($N, M \leq 10^6$).

Формат выходного файла

В выходной файл выведите YES, если числа N и M дружественные, и NO в противном случае.

Примеры

<code>friend.in</code>	<code>friend.out</code>
220 284	YES

Задача В. Шестеренки

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

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

Формат входного файла

Входной файл содержит два целых числа N и K ($1 \leq N, K \leq 2 * 10^9$).

Формат выходного файла

В выходной файл выведите искомое количество зубчиков. Гарантируется, что оно не более 10^9 .

Примеры

<code>gears.in</code>	<code>gears.out</code>
2 4	4
4 6	12

Note

Запрещается использовать типы `int64` и `long long`.

Задача С. Шестеренки - 2

Имя входного файла: `gears2.in`
Имя выходного файла: `gears2.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

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

Формат входного файла

Входной файл содержит два целых числа N и K ($1 \leq N, K \leq 2 * 10^9$).

Формат выходного файла

В выходной файл выведите искомое количество зубчиков.

Примеры

<code>gears2.in</code>	<code>gears2.out</code>
2 4	4
4 6	12

Задача D. Полярные единички

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

Программист на Северном полюсе работал за компьютером в варежках и поэтому мог набирать только 0 и 1, а клавиша 0 запала. Сможет ли он набрать число, состоящее только из единиц и при этом кратное заданному N ?

Формат входного файла

Программе дано число N ($1 \leq N \leq 10^6$).

Формат выходного файла

Вывести минимальное число, удовлетворяющее требованию, или «NO», если такого числа не существует.

Примеры

<code>polar.in</code>	<code>polar.out</code>
100	NO

Задача Е. Разложение на простые

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

Требуется разложить целое число N на простые множители.

Формат входного файла

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

Формат выходного файла

Вывести разложение N на простые множители (см. примеры). Результат требуется выводить в порядке увеличения множителей.

Примеры

<code>primes.in</code>	<code>primes.out</code>
1008	$2^4 * 3^2 * 7$

Note

В формате выходных данных символ « \wedge » обозначает возведение в степень, а « $*$ » — умножение. Например, во втором случае, число 1008 представляется как $2^4 * 3^2 * 7$.

Задача F. МегаНОД

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

Найдите НОД N заданных чисел.

Формат входного файла

Первая строка входного файла содержит натуральное число N ($1 \leq N \leq 1000$) — количество чисел. Во второй строке заданы N чисел, не превышающие по модулю 10^9 .

Формат выходного файла

Выведите НОД N чисел.

Примеры

<code>megagcd.in</code>	<code>megagcd.out</code>
2 90 35	5