

Задача А. Часовая башня

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Многие ученики Хогвартса разозлились на то, что Амбридж запретила собираться в группы из трёх и более учеников, и решили придумать, как ей насолить. Один из планов состоял в том, чтобы прийти в Часовую башню и сломать имеющиеся там часы.

Часы состоят из N последовательно соединённых шестерней, то есть первая шестерня зацепляет вторую, вторая зацепляет третью и так далее... $(N - 1)$ -я шестерня зацепляет последнюю N -ю.

Перед тем как сломать часы, ученики сначала хотят разобраться, как они работают. Им интересно, сколько раз нужно полностью повернуть первую шестерню, чтобы все остальные шестерни вернулись в изначальное положение.

Формат входных данных

В первой строке вводится целое число N ($2 \leq N \leq 10$). В следующей строке вводится N целых чисел a_i — число зубчиков в каждой шестерне ($1 \leq a_i \leq 10^9$).

Формат выходных данных

Выведите минимальное число полных поворотов первой шестерни, после которой все остальные шестерни тоже возвращаются в изначальное положение. Гарантируется, что произведение первого числа и ответа не превышает 10^9 .

Примеры

стандартный ввод	стандартный вывод
3 10 20 30	6

Задача В. Диофантово уравнение

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайта

Даны натуральные числа a , b и c . Решите в целых числах уравнение $ax+by=c$. Среди множества решений следует выбрать такое, где x имеет наименьшее неотрицательное значение.

Формат входных данных

Входной файл содержит три целых числа a и b и c ($1 \leq a, b, c \leq 10^9$).

Формат выходных данных

В выходной файл выведите искомые x и y через пробел. Если решения не существует, выведите одну строку «Impossible».

Примеры

<code>stdin</code>	<code>stdout</code>
1 2 3	1 1

Задача С. Китайская теорема

Имя входного файла: `chine.in`
Имя выходного файла: `chine.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайта

Решите в целых числах систему уравнений

$$\begin{cases} x \equiv a \pmod{n} \\ x \equiv b \pmod{m}, \end{cases}$$

где n и m взаимно просты. Среди решений следует выбрать наименьшее неотрицательное число.

Формат входных данных

Первая строка входных данных содержит число N , $1 \leq N \leq 10^4$, — количество тестов, для которых нужно решить задачу.

Следующие N строк содержат по четыре целых числа a_i, b_i, n_i и m_i ($1 \leq n_i, m_i \leq 10^6$, $0 \leq a_i < n_i$, $0 \leq b_i < m_i$).

Формат выходных данных

Для каждого из тестов выведите искомое наименьшее неотрицательное число x_i .

Примеры

<code>chine.in</code>	<code>chine.out</code>
2	3
1 0 2 3	38
3 2 5 9	

Задача D. Вычислите функции

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

Дано число N . Требуется вычислить следующие функции для него:

$\varphi(N)$ = количество взаимно простых с N чисел среди $1, 2, \dots, N$

$\tau(N)$ = количество делителей числа N

$\sigma(N)$ = сумма всех делителей числа N

Формат входных данных

Во входном файле содержится единственное число $1 \leq N \leq 10^9$.

Формат выходных данных

В единственную строку выходного файла выведите через пробел три числа — значения $\varphi(N)$, $\tau(N)$, $\sigma(N)$.

Примеры

<code>func.in</code>	<code>func.out</code>
2	1 2 3

Задача Е. Обратное по модулю

Имя входного файла: `inverse2.in`
Имя выходного файла: `inverse2.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Даны два целых числа — a, m ($0 \leq a < m$). Нужно найти такое целое x , что $a \cdot x \equiv 1 \pmod{m}$.

Формат входных данных

На первой строке два целых числа — a, m ($0 \leq a \leq 10^{18}, 1 < m \leq 10^{18}, a < m$).

Формат выходных данных

Если такого x не существует, выведите -1 . Иначе выведите целое x ($0 \leq x < m$). Если ответов несколько, выведите любой.

Примеры

<code>inverse2.in</code>	<code>inverse2.out</code>
7 30	13

Задача F. Переливание

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

На досуге вы любите почитать сборники занимательных задач по математике. Недавно вы наткнулись в одном из таких сборников на следующую задачу:

Есть бесконечный резервуар с водой и два пустых сосуда объёмом 5 и 12 литров. Можно: наливать воду из резервуара в любой сосуд до его заполнения, переливать воду из одного сосуда в другой до заполнения второго или опустошения первого (смотря что будет раньше) и выливать воду из сосуда на землю до полного опустошения сосуда. Как таким образом можно отмерить 3 литра?

Вы решили написать программу, которая будет решать подобные задачи для произвольных объёмов сосудов.

Формат входных данных

Во входном файле находятся три целых числа — V_1 , V_2 и V — объёмы двух сосудов и объём воды, который нужно отмерить. Гарантируется, что $1 \leq V_1, V_2 \leq 32767$ и $0 \leq V \leq \max(V_1, V_2)$.

Формат выходных данных

В первую строку выходного файла выведите одно число — количество действий в вашем решении. Далее выведите соответствующее количество строк, описывающих действия в вашем решении. Для каждого действия выведите два числа:

- Если это действие — переливание из одного сосуда в другой, то первое число должно быть номером сосуда, откуда надо переливать воду, а второе — номером сосуда, куда переливать;
- Если это действие — набор воды из резервуара, то первое число должно быть нулём, а второе — номером сосуда, куда наливать;
- Если это действие — выливание воды «на землю», то первое число должно быть номером сосуда, а второе — нулём.

После выполнения всех операций хотя бы в одном сосуде должна находиться вода в объёме V .

Если существует несколько решений, то вы можете вывести любое. Ваше решение не обязано быть оптимальным, единственное ограничение — размер выходного файла не должен превосходить 3 Мб.

Если решений не существует, выведите одно число -1.

Примеры

<code>flow.in</code>	<code>flow.out</code>
120 5 5	2 0 2 2 1
32001 4 32001	1 0 1