

Задача А. Шестерёнки

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
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 входных данных

В единственной строке — два числа, N и K . $1 \leq N, K \leq 10^7$.

Формат выходных данных

Выведите искомое количество зубчиков.

Примеры

<code>stdin</code>	<code>stdout</code>
2 3	6
6 21	42

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. Ну Поллард, и Поллард...

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

Вам необходимо разложить число n на простые множители. Сегодня ночью вам приснился Джон Поллард и сообщил вам половину простых делителей числа n . Формально, если $n = p_0 \times \dots \times p_k$, где $p_0 \leq \dots \leq p_k$, то Поллард сообщил вам числа p_0, p_2, \dots .

Пользуясь информацией, которую сообщил вам Поллард, найти полную факторизацию числа.

Формат входных данных

Первая строка входа содержит два целых числа — число n и l — количество делителей, которые вам сообщил Поллард ($2 \leq n \leq 10^{18}$). Вторая строка содержит l простых чисел — p_0, p_2, \dots .

Формат выходных данных

Выведите простые делители n в порядке неубывания. Количество выводить не нужно.

Примеры

<code>factor.in</code>	<code>factor.out</code>
12 2 2 3	2 2 3
7 1 7	7
1764 3 2 3 7	2 2 3 3 7 7

Задача D. Факторизация

Имя входного файла: pollard.in
Имя выходного файла: pollard.out
Ограничение по времени: 3 секунды
Ограничение по памяти: 256 мегабайт

Дано натуральное число. Факторизуйте его, то есть представьте в виде произведения набора простых чисел. Число p называется простым, если имеет ровно два различных натуральных делителя: 1 и p .

Формат входных данных

В единственной строке записано единственное натуральное число N . $2 \leq N \leq 9 \cdot 10^{18}$.

Формат выходных данных

Выведите в неубывающем порядке одно или несколько простых чисел, произведение которых равно N .

Примеры

pollard.in	pollard.out
6	2 3
7	7

Задача Е. Простые сложности

Имя входного файла: `again.in`
Имя выходного файла: `again.out`
Ограничение по времени: 5 секунды
Ограничение по памяти: 256 мегабайта

В этой жизни не всё так просто. Особенно числа. Вам дан набор чисел. Необходимо для каждого из них определить, является ли оно простым.

Формат входных данных

В первой строке входных данных содержится единственное число $1 \leq T \leq 5\,000$ — количество чисел, которые необходимо проверить на простоту. Далее содержится T целых положительных чисел, не превосходящих 10^{18} .

Формат выходных данных

В i -й строке выходных данных должно быть записано «YES», если i -е число является простым, и «NO» в противном случае.

Примеры

	<code>again.in</code>	<code>again.out</code>
2		YES
3		NO
4		

Задача F. 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 G. RSA. Взлом RSA

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

В 1977 году Ronald Linn Rivest, Adi Shamir и Leonard Adleman предложили новую криптографическую схему RSA, используемую до сих пор. RSA является криптосистемой с открытым ключом: зашифровать сообщение может кто угодно, знающий общеизвестный открытый ключ, а расшифровать сообщение — только тот, кто знает специальный секретный ключ.

Желающий использовать систему RSA для получения сообщений должен сгенерировать два простых числа p и q , вычислить $n = pq$ и сгенерировать два числа e и d такие, что $ed \equiv 1 \pmod{(p-1)(q-1)}$ (заметим, что $(p-1)(q-1) = \varphi(n)$). Числа n и e составляют открытый ключ и являются общеизвестными. Число d является секретным ключом, также необходимо хранить в тайне и разложение числа n на простые множители, так как это позволяет вычислить секретный ключ d .

Сообщениями в системе RSA являются числа из \mathbb{Z}_n . Пусть M — исходное сообщение. Для его шифрования вычисляется значение $C = M^e \pmod n$ (для этого необходимо только знание открытого ключа). Полученное зашифрованное сообщение C передается по каналу связи. Для его расшифровки необходимо вычислить значение $M = C^d \pmod n$, а для этого необходимо знание секретного ключа.

Вы перехватили зашифрованное сообщение C и знаете только открытый ключ: числа n и e . “Взломайте” RSA — расшифруйте сообщение на основе только этих данных.

Формат входных данных

Программа получает на вход три натуральных числа: n , e , C , $n \leq 10^9$, $e \leq 10^9$, $C < n$. Числа n и e являются частью какой-то реальной схемы RSA, т.е. n является произведением двух простых и e взаимно просто с $\varphi(n)$. Число C является результатом шифрования некоторого сообщения M .

Формат выходных данных

Выведите одно число M ($0 \leq M < n$), которое было зашифровано такой криптосхемой.

Примеры

<code>rsa.in</code>	<code>rsa.out</code>
143 113 41	123
9173503 3 4051753	111111

Задача Н. Прямоугольный забег

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Жители Зожбурга очень любят спорт и в особенности бег. Бегать обычные марафоны им надоело, поэтому они решили организовать прямоугольный забег в стиле Minecraft. Для этого на центральной площади города оборудовали стадион с прямоугольным газоном и дорожками вокруг него. Жители Зожбурга считают, что главное — не победа, а участие, поэтому цель забега — сделать красивую фотографию, а не пробежать быстрее всех.

Центральная площадь Зожбурга представляет собой прямоугольник, разделенный на одинаковые единичные квадраты. Строки пронумерованы сверху вниз с единицы, столбцы слева направо с единицы. Каждый квадрат площади имеет координаты r и c — номер строки и столбца, соответственно.

На площади находится прямоугольный газон со сторонами, параллельными сторонам площади. Координаты левого верхнего углового квадрата газона (R_L, C_L) , координаты правого нижнего углового квадрата газона (R_R, C_R) . Вокруг газона оборудованы n дорожек для n бегунов. Дорожка i находится на расстоянии i от границы газона, на дорожке i находится бегун с номером i . Бегун i стартует с квадрата с координатами (r_i, c_i) . Бегуны стартуют одновременно с одинаковой скоростью: через каждую секунду каждый спортсмен меняет текущий квадрат на своей дорожке на следующий квадрат на своей дорожке в направлении против часовой стрелки.

На прямоугольном газоне в квадрате (R_p, C_p) стоит фотограф, цель которого — сделать красивую фотографию. Фотограф тестирует инновационную камеру с двойным объективом. Эта камера делает снимок одновременно в двух противоположных направлениях. Фотограф считает фотографию красивой, если все бегуны в момент, когда он делает снимок, находятся в одновременно в строке R_p или в столбце C_p . При этом благодаря инновационному свойству камеры они могут быть либо в одной строке с ним и справа и слева от него, либо в одном столбце с фотографом и выше и ниже него.

Ваша задача — узнать, через какое минимальное количество секунд t после старта забега фотограф сможет сделать красивую фотографию, или сказать, что красивая фотография в данных условиях не получится.

Формат входных данных

В первой строке входных данных находится число n ($1 \leq n \leq 18$) — количество бегунов. В следующей строке ввода даны шесть целых чисел R_L, C_L, R_R, C_R ($n + 1 \leq R_L \leq R_R \leq 100 - n$, $n + 1 \leq C_L \leq C_R \leq 100 - n$), R_p ($R_L \leq R_p \leq R_R$), C_p ($C_L \leq C_p \leq C_R$) — координаты левого верхнего квадрата газона, правого нижнего квадрата газона, координаты фотографа, соответственно. Гарантируется, что $R_R - R_L + C_R - C_L$ делится на 4.

В следующих n строках даны два числа r_i, c_i — стартовые координаты бегуна i . Гарантируется, что стартовые координаты бегуна i находятся на дорожке i , на каждой дорожке находится один бегун, дорожка i находится на расстоянии i от границы газона.

Формат выходных данных

Выведите единственное число t — через какое минимальное количество секунд t после старта забега фотограф сможет сделать красивую фотографию, или -1 , если фотографию сделать не получится.

Примеры

стандартный ввод	стандартный вывод
2 3 3 5 5 4 3 2 3 1 1	3
2 4 4 5 7 5 4 3 4 7 8	3

Замечание

Рисунок ко второму примеру.

Стартовое положение бегунов.

Положение бегунов через 3 секунды. Все бегуны находятся
в строке R_p , и фотограф делает красивое фото.