Задача A. Задача Альхазена

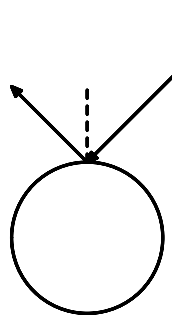
Имя входного файла:	`alhazen.in`
Имя выходного файла:	`alhazen.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

Задача Альхазена об отражении света от зеркальной сферы является одной из классических задач геометрической оптики. Эта задача была сформулирована Ибн ал-Хайсамом, известным европейцам как Альхазен, в фундаментальном трактате «Сокровище оптики». Этот труд состоит из семи книг, три из которых посвящены глазу и зрению. Особый интерес представляет последняя книга: она трактует вопросы преломления света в прозрачных средах.

Альхазен внес существенные уточнения в закон отражения, проверявшийся им на зеркалах, сделанных из железа. Им было установлено, что падающий на поверхность зеркала луч, нормаль к этой поверхности и луч отраженный лежат в одной плоскости.

На основании этого закона задачу об отражении света от сферы можно свести к двумерному случаю. Задана окружность радиуса R с центром в начале координат, от которой происходит отражение луча, положение источника света и точка, через которую прошел луч после отражения от окружности.

Необходимо найти точку, в которой произошло отражение луча от окружности.



Формат входного файла

Первая строка входного файла содержит число наборов n входных данных ($1 \le n \le 50$). Далее следуют описания этих наборов.

Первая строка каждого описания содержит целое число R ($1 \le R \le 1000$). Вторая строка каждого описания содержит два целых числа x_1 и y_1 — координаты источника света ($|x_1| \le 10^4$, $|y_1| \le 10^4$). Третья строка каждого описания содержит два целых числа x_2 и y_2 — координаты точки, через которую прошел луч после отражения ($|x_2| \le 10^4$, $|y_2| \le 10^4$). Обе точки находятся строго вне окружности.

Формат выходного файла

Для каждого набора входных данных в выходной файл выведите координаты точки, в которой произошло отражение луча от окружности. Гарантируется, что входные данные таковы, что такая точка всегда существует. Обе координаты будут сравниваться с точностью до 10^{-3}.

Примеры

alhazen.in	alhazen.out
2 1 2 3 -1 2 2 3 2 0 3	0.0 1.0 0.84105173 1.81456109

Задача B. Отрезок и окружности

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

На плоскости задана система концентрических окружностей, центры которых находятся в начале координат, а радиусы равны 1, 2, 3, ... Также на плоскости задан отрезок, концы которого находятся в точках (x_1, y_1) и (x_2, y_2).

Необходимо найти число общих точек этого отрезка и указанной системы окружностей.

Формат входного файла

Первая строка входного файла содержит четыре целых числа: x_1, y_1, x_2 и y_2. Эти числа не превосходят 10^9 по абсолютной величине. Заданный отрезок имеет ненулевую длину.

Формат выходного файла

В выходной файл выведите ответ на задачу.

Примеры

circles.in	circles.out
1 1 2 1	1
1 2 2 1	0

Задача C. Задача Иосифа

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

Задача Иосифа — это классическая задача информатики и математики. Формулируется она следующим образом. По кругу стоят n людей, они пронумерованы числами от 1 до n по часовой стрелке. Первый из них, начиная с себя, отсчитывает по часовой стрелке k человек, и последний из этих k выходит из круга. После этого счет продолжается, начиная со следующего после выбывшего. Так продолжается до тех пор, пока в круге не останется ровно один человек. Задача состоит в том, чтобы определить его номер человека.

Ваша задача состоит в том, чтобы решить эту задачу для $k = 2$. Напишите программу, которая по числу n определит номер последнего оставшегося в круге человека.

Формат входного файла

Первая строка входного файла содержит целое число n ($2 \le n \le 10^9$).

Формат выходного файла

В выходной файл выведите номер последнего оставшегося в круге человека.

Примеры

josephus.in	josephus.out
3	3
6	5
8	1

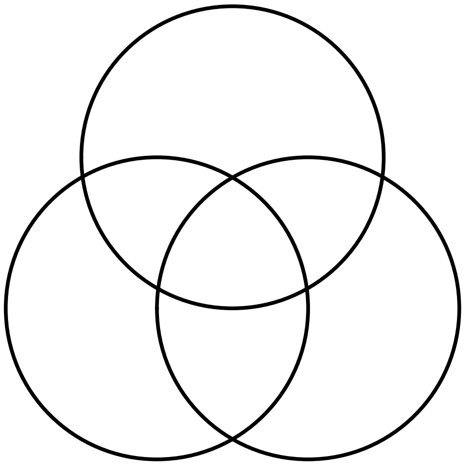
Задача D. Окружности-2

Имя входного файла:	`circles2.in`
Имя выходного файла:	`circles2.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

У Антона есть n колец. Он настолько силен, что может сжимать и растягивать кольца так, чтобы они становились любого радиуса. Каждое кольцо имеет пренебрежимо малую толщину, поэтому его можно рассматривать как окружность.

Теперь Антону интересно, на какое максимальное количество частей он сможет разбить плоскость, если он разложит на ней n колец.

На рисунке изображено разбиение плоскости на восемь частей с помощью трех окружностей.



Формат входного файла

Во входном файле находится одно целое неотрицательное число $n \leq 10^8$.

Формат выходного файла

В выходной файл выведите одно число — максимальное возможное число частей.

Примеры

circles2.in	circles2.out
0	1
1	2
2	4
3	8
4	14

Задача E. Цепная дробь

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

Цепная дробь — это выражение следующего вида.

$$a_0 + \frac{1}{a_1 + \frac{1}{a_2 + \frac{1}{a_3 + \ldots}}}$$

В этом выражении a_0 является целым числом, а остальные a_n — положительными целыми числами. Цепные дроби интересны тем, что с их помощью может быть записано любое вещественное число. При этом для рациональных чисел дробь будет конечной, а для иррациональных — бесконечной.

Например, для числа $\frac{9}{4}$ представление в виде цепной дроби таково: $\frac{9}{4} = 2 + \frac{1}{3+\frac{1}{1}}$.

Ваша задача состоит в том, чтобы найти представление в виде цепной дроби для заданного рационального числа $\frac{p}{q}$.

Формат входного файла

Первая строка входного файла содержит два целых числа: p и q ($1 \le p, q \le 10^3$).

Формат выходного файла

В первой строке выходного файла выведите число n элементов цепной дроби, которая равна $\frac{p}{q}$. Во второй строке выведите числа a_0, a_1, $\ldots$, a_n. Если вариантов представления числа $\frac{p}{q}$ несколько, то выведите любой из них.

Примеры

frac.in	frac.out
9 4	3 2 3 1

Задача F. Двоичное число

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

> Планета Шелезяка...Полезных ископаемых на планете нет. Воды тоже нет. Атмосферы нет. Ничего на планете нет. Если что и было, роботы все истратили и живут в бедности
>
> Кир Булычев «Путешествие Алисы»

Как известно, планета Шелезяка расположена в отдаленной части нашей Галактики, в стороне от космических трасс. Воду роботы (единственные обитатели планеты во второй половине 21-го века) ликвидировали специально, чтобы не ржаветь (они опасались ливней и наводнений). В связи с этим планета оказалась решительно никому кроме роботов не нужна.

Однажды на пустынную планету прилетели пираты. То были настоящие матерые злодеи. Их сердцу была неведома жалость. Они захватили почти всех роботов и увезли их в далекую-далекую галактику и заставили работать на шахтах на планете Бендомир.

Прошло немало времени, прежде чем про это стало известно на Земле. Ричард Темпест, бесстрашный сотрудник института Времени, отправился в прошлое с целью помешать дьявольским планам пиратов. Как оказалось, пользуясь гостеприимством роботов, пираты предлагали роботам решить задачку на скорость. Однако вычислительные возможности роботов были не велики, а оптимизировать алгоритмы их работы было некому. Поэтому все, что оставалось злодеям — загрузить в трюм корабля бесчувственные тела роботов, поглощенных вычислениями.

Ричарду удалось узнать, над чем безуспешно бились электронные мозги роботов. Пираты предлагали им целое неотрицательное число n. После этого они просили узнать, что произойдет с этим числом, если к нему n раз применить операцию: взять старший бит в двоичной записи числа и поставить его в конец. Например, если один раз применить эту операцию к числу $14_{10} = 1110_2$, получится $13_{10} = 1101_2$. После повторного применения получится $11_{10} = 1011_2$.

Ричард присвистнул. Если ему удастся раздобыть программу, которая быстро решает эту задачу, он сможет отправиться во времена до прилета пиратов и загрузить в роботов эту программу, чтобы они смогли противостоять злобным пиратам.

Вы — сотрудник института Времени, которому поручено в срочном порядке написать эту программу. Помогите спасти роботов.

Формат входного файла

В первой строке входного файла содержится целое число n ($0 \le n \le 10^{18}$).

Формат выходного файла

В выходной файл выведите ответ на задачу, поставленную пиратами перед роботами.

Примеры

number.in	number.out
0	0
14	7

Задача G. Точная степень

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

Натуральное число x называется *точной степенью*, если существуют такие натуральные числа a и b ($b > 1$), что $x = a^b$. Заметим, однако, что указанные числа a и b могут определяться неоднозначно. Например, $16 = 2^4 = 4^2$, то есть для $x = 16$ существуют две такие пары (a, b).

Ваша задача состоит в том, чтобы найти все такие пары (a_i, b_i), что $x = a_i^{b_i}$.

Формат входного файла

Входной файл содержит целое число x ($2 \le x \le 10^{18}$).

Формат выходного файла

В первой строке выходного файла выведите число k искомых пар $(a,\ b)$. В каждой из последующих k строк выведите два числа: a_i и b_i.

Примеры

power.in	power.out
3	0
16	2 2 4 4 2

Задача H. Федеральные дороги

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

В стране N-мерике n городов, некоторые из которых соединены федеральными дорогами. Города в стране пронумерованы натуральными числами от 1 до n. Всего в стране m федеральных дорог, каждая из которых соединяет два города. При этом по каждой дороге разрешено движение в обе стороны, а между любые два города соединены не более, чем одной дорогой.

В рамках приоритетного национального проекта «Дороги» решено было построить дополнительные дороги. От различных k строительных компаний в правительственный комитет, занимающийся реализацией этого проекта, поступили заявки. Заявка от компании содержит два номера городов u_i и v_i, между которым эта компания может построить новую федеральную дорогу, длина этой дороги l_i и цена c_i, которую компания требует за постройку этой дороги.

Одной из целей федеральной программы «Дороги» является обеспечение возможности быстрого проезда между некоторыми городами — задано несколько требований, каждое из которых имеет вид: «кратчайший путь между городом a_i и b_i должен быть не длиннее c_i».

Задача конкурсной комиссии состоит в том, чтобы выбрать некоторые из заявок. При этом заявки должны быть выбраны так, чтобы после строительства всех дорог, соответствующих выбранным заявкам, были выполнены все требования. При этом в соответствии с новой программой по борьбе с коррупцией максимальная цена выбранной заявки должны быть как можно меньше.

Необходимо написать программу, которая по описанию дорожной сети N-мерики, описанию поданных заявок на строительство дорог и описанию требований найдет набор заявок, удовлетворяющий поставленным требованиям.

Формат входного файла

Первая строка входного файла содержит число n городов в N-мерике ($1 \le n \le 100$) и число m уже существующих федеральных дорог ($0 \le m \le 10000$). Каждая из последующих m строк описывает одну такую дорогу и содержит три целых числа: u, v, l, где u и v — номера городов, соединенных этой дорогой, а l — ее длина ($1 \le u, v \le n$, $u \ne v$, $1 \le l \le 10^4$).

Следующая строка входного файла содержит число k заявок на строительство дорог, поступивших от строительных фирм ($1 \le k \le 10000$). Последующие k строк описывают эти заявки, каждая из них содержит четыре числа u, v, l, c, где u и v — номера городов, которые будут соединены дорогой ($1 \le u, v \le n$, $u \ne v$), l — длина этой дороги ($1 \le l \le 10^4$), а c — стоимость ее строительства ($1 \le c \le 10^9$). Ни в одной из заявок не предлагается соединить дорогой два города, между которыми дорога уже есть.

Следующая строка входного файла содержит число p требований к дорожной сети ($1 \le p \le 1000$). Последующие p строк описывают эти требования – каждая из них содержит три целых числа a, b, c ($1 \le a, b \le n$, $a \ne b$, $1 \le c \le 10^8$). Такая строка описывает требование «кратчайший путь между городом a и b должен быть не длиннее c».

Формат выходного файла

В первой строке выходного файла выведите число заявок, которые должны удовлетворены, для того чтобы были выполнены все требования к дорожной сети. При это максимальная стоимость удовлетворенной заявки должны быть минимальной возможной. Во второй строке выведите в возрастающем порядке номера этих заявок.

Если требования к дорожной сети не могут быть выполнены, выведите в выходной файл одно число `-1`.

Примеры

roads.in	roads.out
3 0 3 1 2 2 1 2 3 2 1 1 3 1 3 1 1 3 4	2 1 2
3 1 1 2 2 2 2 3 2 1 1 3 1 3 1 1 3 3	2 1 2

Задача I. Число возрастающих подпоследовательностей

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

Задана последовательность из n чисел a_1, a_2, …, a_n. *Подпоследовательностью* длины k этой последовательности называется набор индексов i_1, i_2, …, i_k, удовлетворяющий неравенствам $1 \le i_1 < i_2 < \ldots < i_k \le n$. Подпоследовательность называется *возрастающей*, если выполняются неравенства $a_{i_1} < a_{i_2} < \ldots < a_{i_k}$.

Необходимо найти число возрастающих подпоследовательностей наибольшей длины заданной последовательности a_1, …, a_n. Так как это число может быть достаточно большим, необходимо найти остаток от его деления на $10^9 + 7$.

Формат входного файла

Первая строка входного файла содержит целое число n ($1 \le n \le 100000$). Вторая строка входного файла содержит n целых чисел: a_1, a_2, …, a_n. Все a_i не превосходят 10^9 по абсолютной величине.

Формат выходного файла

В выходной файл выведите ответ на задачу.

Примеры

subseq.in	subseq.out
5 1 2 3 4 5	1
6 1 1 2 2 3 3	8

Задача J. Закупка билетов

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

Школа №932 решила закупить билеты в цирк. Каждый билет имеет k-значный номер, состоящий из цифр от 0 до 9. Билет стоит C рублей. Однако за каждую встречающуюся в номере билета «счастливую» цифру D делается скидка в P процентов.

Более формально, пусть в номере билета есть m цифр D. Тогда цена билета с учетом всех скидок составит $C \cdot (1 - P/100)^m$ рублей (округленная вниз).

Для оптимизации закупок необходимо провести некоторые исследования. В числе их Вам поручено узнать суммарную стоимость билетов с номерами от K_{min} до K_{max} включительно.

Формат входного файла

В первой строке входного файла находится количество интервалов N, для которых нужно узнать стоимость.

В следующей строке находятся три числа — C ($1 \le C \le 10^4$), D ($0 \le D \le 9$), P ($0 \le P \le 100$).

В последующих N строках находятся пары номеров K_{min}, K_{max}. В каждой строке оба номера состоят из цифр от 0 до 9, их длина одинаковая, ненулевая и не превосходит 13. Гарантируется, что первый из этих номеров не превосходит второй.

Формат выходного файла

Для каждого интервала выведите его стоимость. Выводите по одному числу на каждой строке.

Примеры

tickets.in	tickets.out
1 100 1 10 0 9	990
1 100 9 10 00 99	9801