Задача A. Задача по математике

Имя входного файла: input.txt
Имя выходного файла: output.txt
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

Иван Петрович, учитель математики в средней школе города N-ска, хочет провести контрольную работу на тему «делимость целых чисел». Учитель хочет составить очень много различных вариантов контрольной работы, чтобы ученики не могли списать друг у друга.

Иван Петрович придумал следующую задачу: пусть даны целые числа A и N, найдите целое число B такое, что $AB + A + B$ делится на N. Он решил составить несколько подобных задач, изменяя данные в условии задачи. Для того, чтобы быстро проверить решение задачи учениками учителю необходимо знать и ответы на эти задачи. Помогите Ивану Петровичу написать программу, которая будет решать его задачу для различных исходных данных, поможет сэкономить много времени и сил при подготовке и проверке контрольной работы.

Формат входных данных

Единственная строка входного файла содержит два целых числа A, N, разделённых знаком пробела – числа, которые будут использоваться для очередного варианта контрольной ($1 \leq A, N \leq 10^9$).

Формат выходных данных

В выходной файл нужно вывести одно число B – ответ на задачу Ивана Петровича. Число B должно быть неотрицательным и не превышать 10^9. Если такого B не найдется – необходимо вывести число -1.

Примеры

input.txt	output.txt
2 4	2
3 4	-1

Разбор

По условию задачи нужно найти такое число B, чтобы $AB+A+B$ делилось на N. Заметим, что $AB+A+B=(A+1)(B+1)-1$. Таким образом, произведение A+1 и B+1 дает остаток 1 при делении на N. В таком случае говорят, что остаток числа B+1 при делении на N является обратным для остатка A+1. Известно, что обратный остаток существует тогда и только тогда, когда A+1 взаимно простое с N.

Для нахождения обратного остатка для данного остатка можно воспользоваться расширенным алгоритмом Евклида. Этот алгоритм позволяет найти наибольший простой делитель чисел A+1 и N, и если они оказываются взаимно просты, то он также находит целые числа X и Y такие, что $(A+1)X+NY=1$. В этом случае остаток X при делении на N будет обратным для остатка A+1.

Задача B. Почти беспрефиксные коды

Имя входного файла: input.txt
Имя выходного файла: output.txt
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

В теории кодирования часто используют *беспрефиксные коды* – наборы слов, ни одно из которых не является префиксом[1] другого. Например, набор слов «aba», «aa» и «bac» является беспрефиксным кодом, а набор «abac», «aba», «ba» – нет, поскольку слово «aba» является префиксом слова «abac».

Профессор главного университета города Соколовка де Фрошин дает задания нерадивым студентам по кодированию информации – *почти беспрефиксные коды*. Набор слов называется почти беспрефиксным кодом уровня k, если наибольший общий префикс двух любых слов из набора не превышает по длине k. Например, набор «abac», «abc», «ba» является почти беспрефиксным кодом уровня 2, а набор «abac», «abab», «ba» – уровня 3, поскольку наибольший общий префикс слов «abac» и «abab» имеет длину 3.

Задача, которую профессор де Фрошин поставил своим студентам, заключается в следующем: по заданному набору слов и числу k требуется выбрать из заданных слов максимальный набор, который является почти беспрефиксным кодом уровня k. А вас он просит написать программу для помощи в проверке ответов.

Формат входного файла

Первая строка входного файла содержит два целых числа: n и k – количество слов в заданном наборе и уровень почти беспрефиксного кода, который требуется построить ($1 \le n \le 100\,000$, $0 \le k \le 200$). Следующие n строк содержат по одному слову. Слова состоят из строчных букв латинского алфавита. Длина каждого слова от 1 до 200 символов. Суммарная длина всех слов не превышает 10^6. Все слова различны.

Формат выходного файла

В первой строке выходного файла выведите одно число m – максимальное количество слов, которые можно выбрать из заданного набора, чтобы они образовывали почти беспрефиксный код уровня k. Следующие m строк должны содержать выбранные слова.

Примеры

input.txt	output.txt
6 2 aba bacaba abacaba baca abac caba	3 aba bacaba caba

Разбор

Почти беспрефиксный код уровня k – набор слов, у которых наибольший общий префикс имеет длину не более k.

Для решения задачи отсортируем все строки в лексикографическом порядке. Далее будем добавлять в искомый набор слово, если его общий префикс с предыдущим не превышает k.

[1] Слово α называется префиксом слова β, если α получается из β удалением нуля или более символов в конце. Например, слова «», «a», «ab» и «aba» являются префиксами слова «aba»

Задача C. Мост

Имя входного файла: input.txt
Имя выходного файла: output.txt
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Город Соколовка находится на берегах реки Флатовка, протекающей с юга на север. Жители города очень деловые и поэтому написали письмо мэру города с просьбой построить новый мост через реку для того, чтобы можно было перемещаться из одной части города в другую без автомобильных пробок. Однако возможности городского бюджета ограничены, поэтому мэр решил построить мост минимально возможной длины. Вдоль реки по обоим берегам проходят автомобильные дороги, поэтому выезд на мост можно сделать в любом месте.

Для упрощения задачи поиска местоположения моста, можно представить, что берега реки представляют собой ломаные, бесконечные в обе стороны. Введем координатную систему таким образом, чтобы ось OY была направлена с юга на север, а ось OX – с запада на восток. Левый берег начинается лучом, направленным на юг из точки $(x_{1,1}, y_{1,1})$, продолжается отрезками

$$(x_{1,1}, y_{1,1}) - (x_{1,2}, y_{1,2}),\ (x_{1,2}, y_{1,2}) - (x_{1,3}, y_{1,3}),\ \ldots,\ (x_{1,m-1}, y_{1,m-1}) - (x_{1,m}, y_{1,m})$$

и заканчивается лучом, направленным на север из точки $(x_{1,m}, y_{1,m})$. Аналогично, правый берег реки начинается лучом, направленным на юг из точки $(x_{2,1}, y_{2,1})$, продолжается отрезками

$$(x_{2,1}, y_{2,1}) - (x_{2,2}, y_{2,2}),\ (x_{2,2}, y_{2,2}) - (x_{2,3}, y_{2,3}),\ \ldots,\ (x_{2,n-1}, y_{2,n-1}) - (x_{2,n}, y_{2,n})$$

и заканчивается лучом, направленным на север из точки $(x_{2,n}, y_{2,n})$.

Помогите мэру города выяснить, мост какой минимальной длины можно построить.

Формат входного файла

Первая строка входного файла содержит целое число m ($2 \le m \le 100$). Следующие m строк содержат по два целых числа – координаты вершин ломаной левого берега: $x_{1,1}; y_{1,1}, x_{1,2}; y_{1,2}, \ldots, x_{1,m}; y_{1,m}$.

Следующая строка входного файла содержит целое число n ($2 \le n \le 100$). Следующие n строк содержат по два целых числа – координаты вершин ломаной правого берега: $x_{2,1}; y_{2,1}, x_{2,2}; y_{2,2}, \ldots, x_{2,n}; y_{2,n}$.

Известно, что $x_{1,1} < x_{2,1}$, каждая из ломаных не имеет самопересечений и самокасаний, ломаные не имеют общих точек. Все отрезки каждой из ломаных имеют положительную длину. Все координаты не превосходят 10^4 по абсолютной величине.

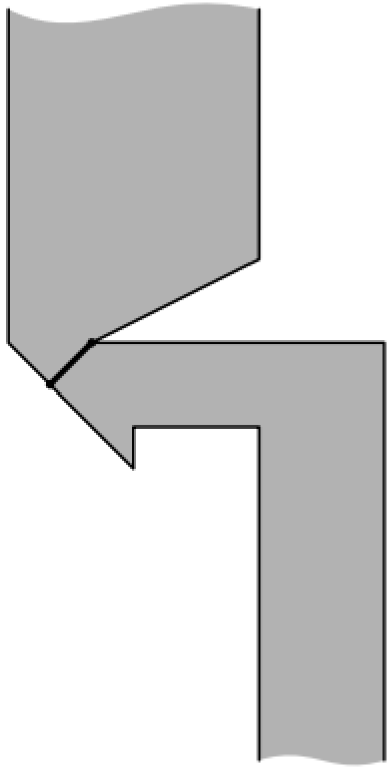
Формат выходного файла

Выведите в выходной файл одно вещественное число: минимальную возможную длину моста. Ваш ответ будет проверяться с точностью 10^{-5}.

Пример

input.txt	output.txt
4 6 1 3 1 3 0 0 3 3 9 3 2 3 6 5	1.41421356237309505

Оптимальное положение моста показано на следующем рисунке:

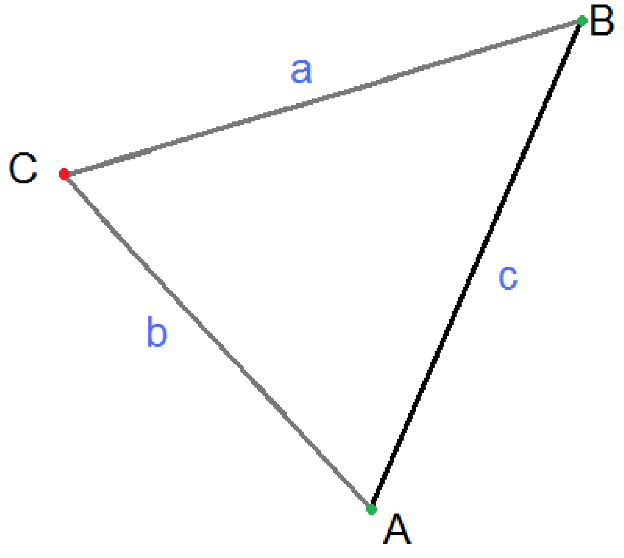


Разбор

Берега реки – ломаные, бесконечные в обе стороны. Требуется построить мост через реку минимальной длины.

Если ответ – L, то всегда существует мост длины L, построенный через вершину ломаной, перпендикулярно одному из отрезков ломанной противоположного берега реки. Достаточно для каждой вершины каждой ломаной проверить расстояние до каждого отрезка и луча второй ломаной.

Поиск расстояния от точки до отрезка. Найдем расстояние от точки C до отрезка AB.



Если $\begin{cases} a^2 + c^2 \geq b^2 \\ b^2 + c^2 \geq a^2 \end{cases}$, то расстояние до отрезка есть расстояние до прямой, содержащей этот отрезок. Иначе, это min(a, b).

Задача D. Половина

Имя входного файла: input.txt
Имя выходного файла: output.txt
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Лешу навестили родственники из деревни и привезли много персиков. Поскольку персиков было действительно много, и они быстро портятся, Леша решил поделиться с друзьями. Каждый раз, когда он встречает какого-либо своего друга, он смотрит на персики, которые у него есть и отдает другу половину.

Но Леша не одинаково любит всех своих друзей, поэтому некоторым из них он отдает половину всех имеющихся у него персиков, а некоторым – только половину персика. При этом он решил не делить фрукты более, чем на две части. Поэтому, если он встречает друга, а у него нецелое или нечетное число персиков, то он вынужден разделить один из фруктов надвое.

Утром у Леши было n персиков, а за день он встретил k друзей. Выясните, сколько персиков у него могло остаться вечером.

Формат входного файла

Входной файл содержит два целых числа: n – количество персиков у Леши и k – количество встреченных им за день друзей ($1 \leq n \leq 1000$, $1 \leq k \leq 1000$).

Формат выходного файла

Первая строка выходного файла должна содержать число m – количество вариантов ответа на вопрос, сколько персиков может быть у Леши вечером. Следующая строка должна содержать m вещественных чисел, отсортированных по возрастанию – варианты ответов.

Примеры

input.txt	output.txt
6 1	2 3.0 5.5

Разбор

Дано N персиков, k раз Леша либо отдает половину от всех своих фруктов, либо половину одного фрукта. Леша не может делить персик более, чем на две части. Найти, сколько персиков могло остаться у Леши.

Воспользуемся методом динамического программирования.

Будем хранить ответы на вопрос: «Может ли у Леши быть X персиков после того, как он встретил Y друзей?».

Заметим, что X лучше хранить в «половинках» персиков, т. к. их количество всегда целое число.

Пусть у Леши может быть X «половинок» персиков после встречи Y друзей. В какие состояния можно перейти?

1. Если у Леши есть хотя бы одна «половинка», он может ее отдать.
2. Если у Леши четное число «половинок», то он может отдать половину от всех персиков.

Значит, из состояния динамики ($X; Y$), где X – количество персиков, Y – количество уже встреченных друзей, будут следующие переходы:

1. Если $X > 0$, тогда $(X - 1; Y + 1)$
2. Если X делится на 2, тогда $(\frac{X}{2}; Y + 1)$

Сложность полученного решения – $O(N \cdot K)$.