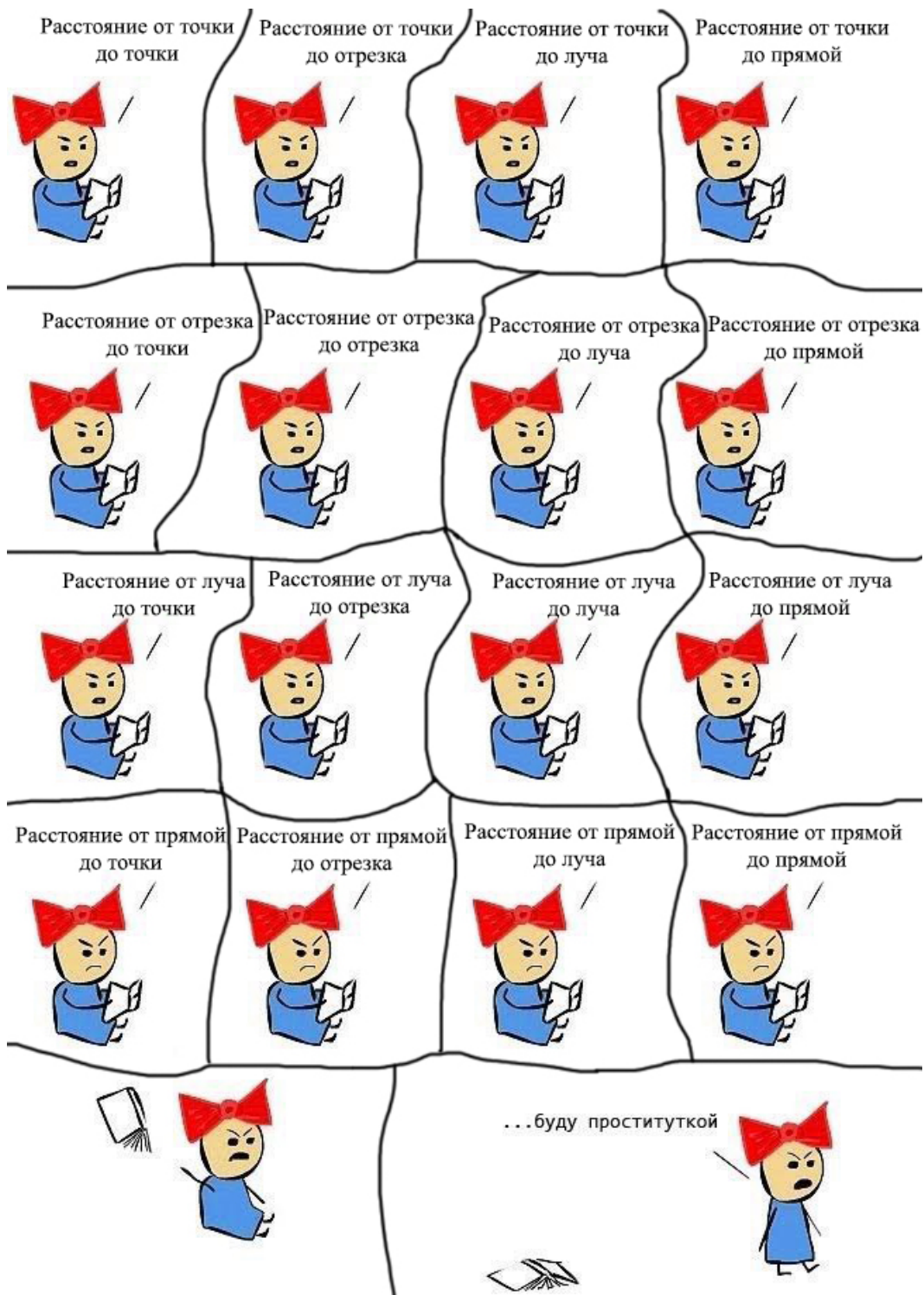


Задача А. 16

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

Даны 4 точки A, B, C, D.
Посчитайте:



Даны 4 точки A, B, C, D.

Посчитайте:

- Расстояние от точки A до точки C.
- Расстояние от точки A до отрезка CD.
- Расстояние от точки A до луча CD.
- Расстояние от точки A до прямой CD.
- Расстояние от отрезка AB до точки C.
- Расстояние от отрезка AB до отрезка CD.
- Расстояние от отрезка AB до луча CD.
- Расстояние от отрезка AB до прямой CD.
- Расстояние от луча AB до точки C.
- Расстояние от луча AB до отрезка CD.
- Расстояние от луча AB до луча CD.
- Расстояние от луча AB до прямой CD.
- Расстояние от прямой AB до точки C.
- Расстояние от прямой AB до отрезка CD.
- Расстояние от прямой AB до луча CD.
- Расстояние от прямой AB до прямой CD.

Формат входных данных

Даны координаты четырех точек, по одной точке в строке: $X_a, Y_a, X_b, Y_b, X_c, Y_c, X_d, Y_d$. Все числа целые, по модулю не превосходят 10000.

Формат выходных данных

Выведите 16 чисел по одному в строке. Числа в ответе должны быть выданы с точностью не менее 6 знаков после десятичной точки.

Пример

стандартный ввод	стандартный вывод
1 2	5.6568542495
7 1	5.6000000000
5 6	5.6000000000
8 2	5.6000000000
	4.6031716446
	1.4142135624
	1.4000000000
	1.4000000000
	4.6031716446
	1.1507929111
	0.0000000000
	0.0000000000
	4.6031716446
	1.1507929111
	0.0000000000
	0.0000000000

Задача В. Извилистая ломаная

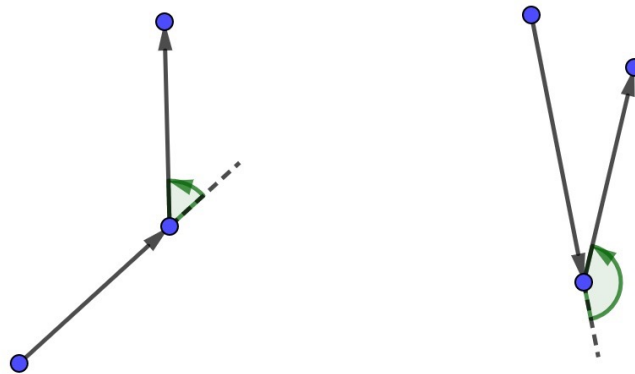
Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

У Васи есть n различных точек A_1, A_2, \dots, A_n на плоскости. Никакие три из них не лежат на одной прямой. Он хочет расположить их в некотором порядке $A_{p_1}, A_{p_2}, \dots, A_{p_n}$, где p_1, p_2, \dots, p_n — это некоторая перестановка чисел от 1 до n .

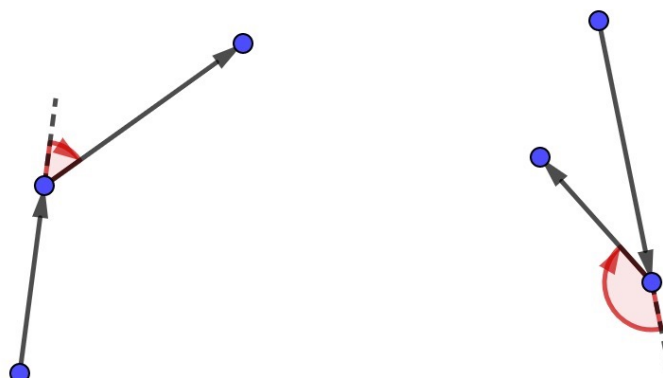
Сделав так, он нарисует ориентированную ломаную на этих вершинах, проведя направленные отрезки из каждой точки в следующую в выбранном порядке точку. То есть для всех $1 \leq i \leq n-1$ он проведет направленный отрезок из точки A_{p_i} в точку $A_{p_{i+1}}$. Он хочет, чтобы получившаяся ломаная удовлетворяла 2-м условиям:

- она будет несамопересекающейся, то есть любые 2 отрезка, которые не являются соседними, не имеют общих точек.
- она будет извилистой.

У Васи есть строка s , состоящая из $(n-2)$ -х символов “L” или “R”. Будем называть направленную ломаную извилистой, если её i -й поворот будет налево, если $s_i = \text{“L”}$ и направо, если $s_i = \text{“R”}$. Более формально: i -й поворот ломаной будет в точке $A_{p_{i+1}}$, в ней направленный отрезок из точки A_{p_i} в точку $A_{p_{i+1}}$ поменяется на направленный отрезок из точки $A_{p_{i+1}}$ в точку $A_{p_{i+2}}$. Обозначим вектор $\vec{v}_1 = \overrightarrow{A_{p_i}A_{p_{i+1}}}$ и вектор $\vec{v}_2 = \overrightarrow{A_{p_{i+1}}A_{p_{i+2}}}$. Тогда если для того, чтобы повернуть вектор \vec{v}_1 на наименьший возможный угол, чтобы его направление совпало с направлением вектора \vec{v}_2 надо сделать поворот против часовой стрелки, то будем говорить, что i -й поворот налево, а иначе направо. Для лучшего понимания посмотрите картинки, на которых изображены различные варианты поворотов:



На этой картинке изображены повороты налево



На этой картинке изображены повороты направо

Вам даны координаты точек A_1, A_2, \dots, A_n на плоскости и строка s . Найдите перестановку p_1, p_2, \dots, p_n чисел от 1 до n , такую что ломаная, которую нарисует Вася, будет удовлетворять двум заданным условиям.

Формат входных данных

В первой строке написано одно целое число n — количество точек ($3 \leq n \leq 2000$). В следующих n строках написаны по два целых числа x_i и y_i , разделённые пробелом — координаты точки A_i на плоскости ($-10^9 \leq x_i, y_i \leq 10^9$). В последней строке написана строка s из символов “L” и “R” длины $(n - 2)$. Гарантируется, что все точки различны и никакие три точки не лежат на одной прямой.

Формат выходных данных

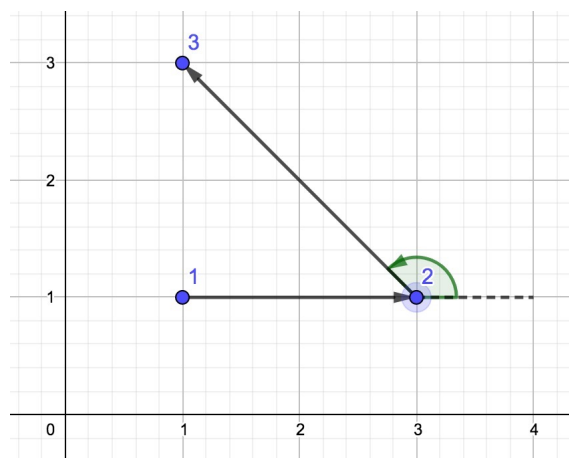
Если подходящей перестановки не существует выведите -1 . Иначе выведите n чисел p_1, p_2, \dots, p_n — найденную перестановку ($1 \leq p_i \leq n$ и все p_1, p_2, \dots, p_n различны). Если подходящих перестановок несколько, выведите любую.

Примеры

стандартный ввод	стандартный вывод
3 1 1 3 1 1 3 L	1 2 3
6 1 0 0 1 0 2 -1 0 -1 -1 2 1 RLLR	5 4 1 6 2 3

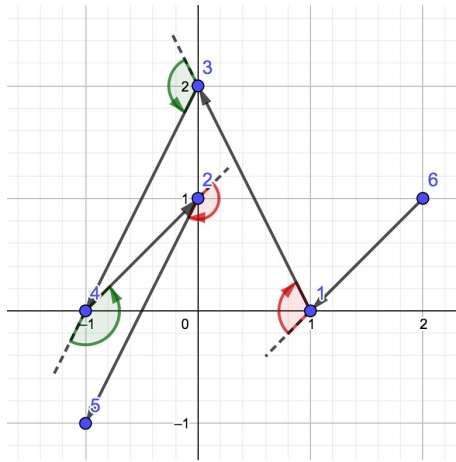
Замечание

Вот картинка, изображающая ломаную из 1 теста:



Как мы видим, эта ломаная несамопересекающаяся, а также извилистая, так как поворот в точке 2 налево.

Вот картинка, изображающая ломаную из 2 теста:



Задача С. U2

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

Недавно Вася узнал, что через любые две точки на плоскости с различными x координатами можно провести одну и только одну параболу, уравнение которой будет иметь вид $y = x^2 + bx + c$, где b и c — действительные числа. Назовём такую параболу U -образной.

После этого он нарисовал на плоскости несколько различных точек с целыми координатами и через каждые две из них, имеющие различные координаты x , провёл U -образную параболу. Рисунок получился откровенно плохой, но Вася не теряет надежды найти число различных получившихся парабол, во внутренней области каждой из которых нет ни одной нарисованной точки. Помогите Васе.

Внутренней областью U -образной параболы в данной задаче называется часть плоскости, лежащая строго выше параболы, при этом ось y направлена вверх.

Формат входных данных

В первой строке находится единственное целое число n ($1 \leq n \leq 100\,000$) — число точек.

В следующих n строках находятся описания точек, в i -й из них находятся 2 целых числа x_i и y_i — координаты i -й точки. Гарантируется, что все точки различные, а так же что координаты по модулю не превосходят 10^6 .

Формат выходных данных

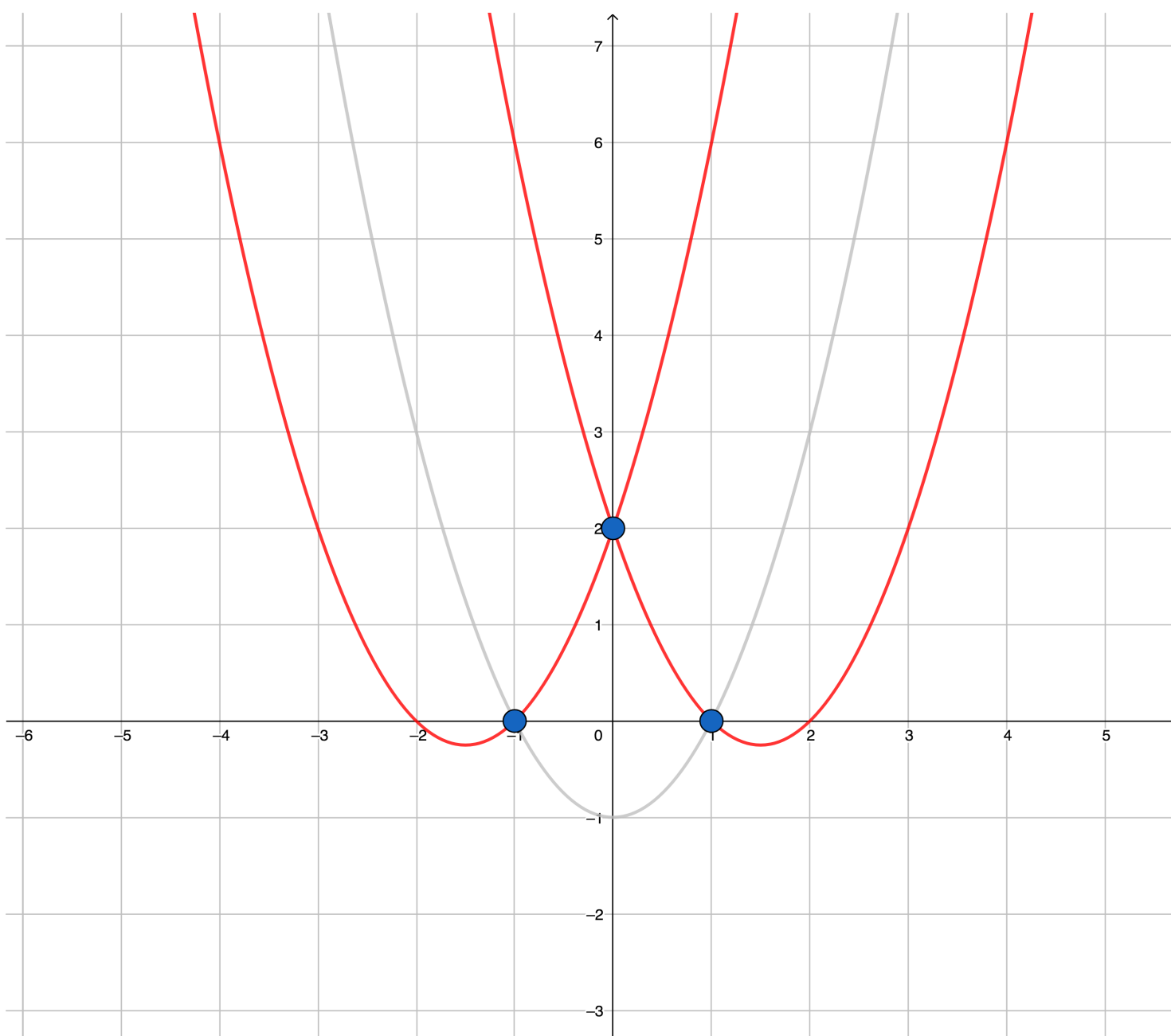
В единственной строке выведите единственное число — количество U -образных парабол, проходящих хотя бы через 2 точки и не содержащих никаких других точек в своей внутренней области (не считая границы).

Примеры

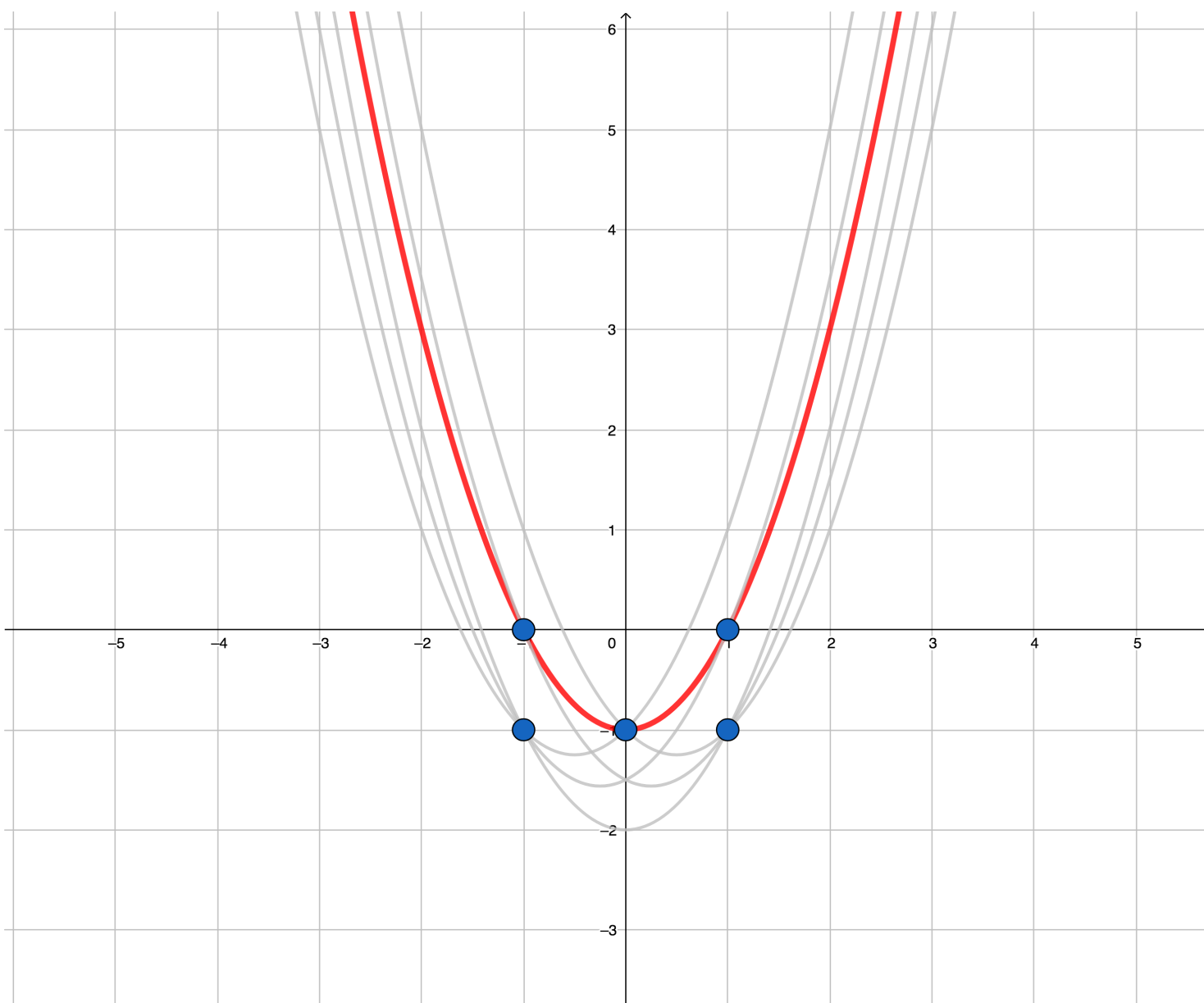
стандартный ввод	стандартный вывод
3 -1 0 0 2 1 0	2
5 1 0 1 -1 0 -1 -1 0 -1 -1	1

Замечание

На двух картинках ниже нарисованы все U -образные параболы, проходящие хотя-бы через 2 отмеченные точки в каждом из двух примеров. Красным отмечены те U -образные параболы, во внутренней области которых нет ни одной точки.



Первый пример.



Второй пример.

Задача D. Теодор Рузвельт

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

«Теодор Рузвельт» — флагман военно-морского флота Кукуляндии. Заклятые враги кукуляндцев, флатландцы, решили уничтожить его. Они узнали, что «Теодор Рузвельт» представляет собой выпуклый многоугольник из n вершин и узнали его координаты. Затем они выпустили m баллистических ракет и определили координаты точек, где эти ракеты взорвались. По расчётам штаба флатландцев, «Теодор Рузвельт» будет уничтожен, если в него попадёт хотя бы k ракет. Вычислите, удалось ли флатландцам уничтожить корабль.

Формат входных данных

В первой строке через пробел записаны целые числа n, m, k ($3 \leq n \leq 10^5, 0 \leq k \leq m \leq 10^5$). В последующих n строках записаны координаты вершин многоугольника в порядке обхода против часовой стрелки. В следующих m строках записаны координаты точек. Гарантируется, что все координаты — целые числа, не превосходящие по модулю 10^9 .

Формат выходных данных

Выведите «YES», если в многоугольнике или на его границе лежит по крайней мере k точек, и «NO» в противном случае.

Пример

стандартный ввод	стандартный вывод
5 4 2 1 -1 1 2 0 4 -1 2 -1 -1 -2 -1 1 -1 0 1 2 3	YES

Задача Е. Адская мухобойка

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

У Жени дома летает очень много ос. Они постоянно летают под потолком в одних и тех же местах. Теперь Евгений отправился в магазин для покупки новой мухобойки. Все мухобойки имеют форму круга с различными радиусами. Женя — очень экономный студент, поэтому он решил купить самую дешёвую мухобойку — с минимально возможным радиусом, но Женя так же очень прагматичен, поэтому он купит только такую мухобойку, что с её помощью можно будет одним ударом убить всех ос. Помогите ему! Для простоты можете считать, что на потолке введена стандартная декартова система координат, и координаты ос постоянны. Помните, что ос у Жени действительно много.

Формат входных данных

В первой строке входного файла содержится число N — количество ос ($1 \leq N \leq 100\,000$). Далее содержатся координаты ос — пара целых чисел, не превосходящих по модулю 10^6 .

Формат выходных данных

В первой строке выходных данных выведите координаты точки, в которой Евгений должен нанести свой сокрушительный удар (это та точка, в которой будет расположен центр мухобойки). На следующей строке выведите одно число — минимальный радиус мухобойки, которого будет достаточно, чтобы уничтожить всех омерзительных ос. Ваш ответ будет считаться правильным, если его абсолютная или относительная погрешность не будет превышать 10^{-6} .

Пример

стандартный ввод	стандартный вывод
3	1.000 1.000
0 2	1.4142135623730951
0 0	
2 0	

Задача F. Well, Just You Wait!

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мегабайт

Во время съемок мультсериала «Ну, погоди!» Васе — актёру, игравшему роль волка, приходилось участвовать в погонях за Петей, игравшим роль зайца. Но никто на съёмочной площадке не подозревал о том, что Вася в самом деле хочет съесть Петю!

Съёмочная площадка представляет собой выпуклый n -угольник на плоскости. Петя очень любит природу, и в каждый из следующих m дней планирует посадить дерево прямо на съёмочной площадке. Вася хочет спрятаться где-то на площадке, чтобы утром, когда Петя придет посадить дерево, выпрыгнуть и съесть его.

Съёмочная площадка — место полное неожиданностей: в любой момент по указанию режиссера монтажники могут построить стену, представляющую собой отрезок, соединяющий две вершины многоугольника. Вася не хочет, чтобы его план сорвался, поэтому он решил, что спрячется в таком месте, что какую бы стену не построили монтажники, она не будет разделять Васю и Петю. (Если вдруг стену проложат прямо через место, где прячется Вася, он может сдвинуться в ту часть многоугольника, где находится Петя).

Вася не хочет, чтобы Петя заметил его до нападения. Поэтому среди всех подходящих мест он хочет выбрать место, находящееся как можно дальше от точки, в которой Петя будет сажать дерево.

Помогите Васе: для каждого из m дней определите наибольшее расстояние, на котором Вася может спрятаться от Пети.

Формат входных данных

В первой строке входного файла содержится число n — число вершин многоугольника, соответствующего съёмочной площадке ($3 \leq n \leq 200$).

Следующие n строк содержат пары целых чисел x_i, y_i — координаты точек, являющихся вершинами многоугольника ($-10\,000 \leq x_i, y_i \leq 10\,000$) в порядке обхода против часовой стрелки. Гарантируется, что многоугольник является выпуклым и никакие три вершины не лежат на одной прямой.

В следующей строке содержится число m — число деревьев, которое собирается посадить Петя ($1 \leq m \leq 200$).

Следующие m строк содержат пары целых чисел u_i, v_i — координаты точки, в которой Петя собирается посадить дерево в i -й день ($-10\,000 \leq u_i, v_i \leq 10\,000$). Гарантируется, что каждая из этих точек лежит строго внутри многоугольника и не лежит на прямой, соединяющей две вершины многоугольника.

Формат выходных данных

Выведите m строк.

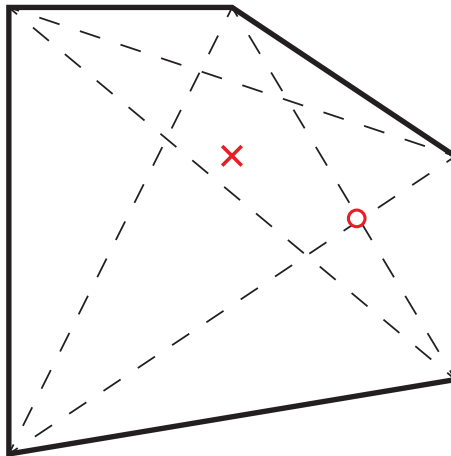
В i -й строке выходного файла выведите одно число — максимальное расстояние, на котором Вася может спрятаться от Пети, когда Петя будет сажать дерево в i -й день. Ваш ответ будет считаться правильным, если его абсолютная или относительная погрешность относительно правильного ответа составит не более 10^{-6} . А именно: пусть ваш ответ равен a , а ответ жюри — b . Проверяющая программа будет считать ваш ответ правильным, если $\frac{|a-b|}{\max(1,a)} \leq 10^{-6}$.

Примеры

стандартный ввод	стандартный вывод
5 5 -2 5 1 2 3 -1 3 -1 -3 1 2 1	1.9166296949998198
3 3 1 10 3 5 7 3 5 2 9 3 6 6	5.0990195135927845 6.324555320336759 5.830951894845301

Замечание

Ниже приведена иллюстрация к первому примеру. Пунктирные линии обозначают возможные положения стены. Крестик обозначает точку, где Петя собирается посадить дерево, кружок — оптимальную позицию для Васи.



Задача G. Клуб анонимных геометров

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мегабайт

Сегодня Денис проводит в ЛКШ клуб анонимных геометров. Он заготовил для клуба n выпуклых многоугольников, пронумерованных от 1 до n . Он планирует предложить участникам клуба посчитать суммы Минковского этих многоугольников. А именно, он планирует дать q заданий, в i -м из них нужно посчитать сумму Минковского многоугольников с номерами от l_i до r_i включительно.

Суммой Минковского двух множеств A и B называется множество $C = \{a + b : a \in A, b \in B\}$. Можно доказать, что если A и B являются выпуклыми многоугольниками, то C тоже будет выпуклым многоугольником.

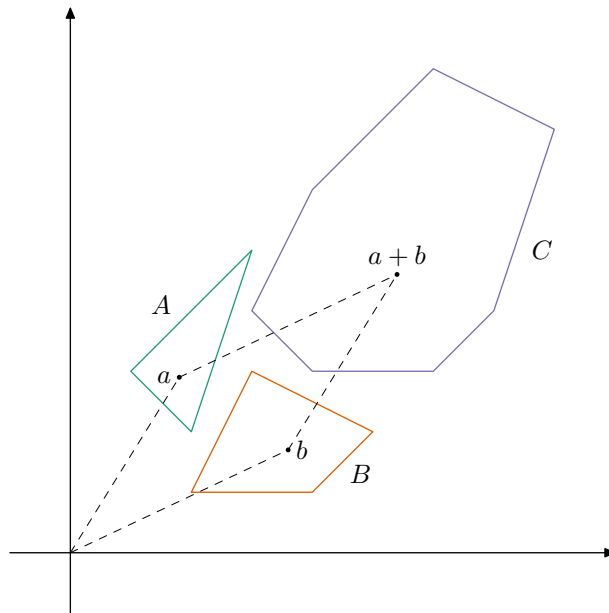


Рис. 1: Сумма двух двух выпуклых многоугольников

Чтобы посчитать сумму Минковского p многоугольников ($p > 2$), нужно посчитать сумму Минковского первых $p - 1$ многоугольников, а потом сумму Минковского результата и p -го многоугольника.

Для того, чтобы было удобнее проверять правильность выполнения заданий, Денис решил подготовиться и посчитать количество вершин в сумме Минковского для каждого задания. Помогите ему сделать это.

Формат входных данных

В первой строке дано одно целое число n — количество выпуклых многоугольников, которые подготовил Денис ($1 \leq n \leq 100\,000$).

Далее дано описание n выпуклых многоугольников. Описание i -го многоугольника начинается с целого числа k_i — количество вершин в i -м многоугольнике ($3 \leq k_i$). После чего в k_i строках даны по два целых числа x_{ij}, y_{ij} — координаты вершин i -го многоугольника в порядке обхода против часовой стрелки ($|x_{ij}|, |y_{ij}| \leq 10^9$).

Гарантируется, что многоугольники не содержат трех последовательных вершин, лежащих на одной прямой. Суммарное количество вершин в многоугольниках не превышает 300 000.

В следующей строке одно целое число q — количество заданий ($1 \leq q \leq 100\,000$). В следующих q строках дано описание заданий. Описание i -го задания содержит два целых числа l_i и r_i ($1 \leq l_i \leq r_i \leq n$).

Формат выходных данных

Для каждого задания выведите на новой строке одно целое число — количество вершин в сумме Минковского многоугольников с номерами от l_i до r_i .

Пример

стандартный ввод	стандартный вывод
3	5
3	5
0 0	6
1 0	
0 1	
4	
1 1	
1 2	
0 2	
0 1	
3	
2 2	
1 2	
2 1	
3	
1 2	
2 3	
1 3	

Замечание

Пояснение к примеру:



(a) 1-й многоугольник

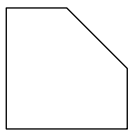


(b) 2-й многоугольник

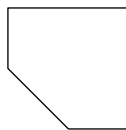


(c) 3-й многоугольник

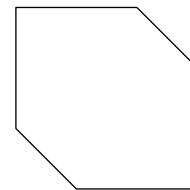
Рис. 2: Многоугольники, которые заготовил Денис



(a) 1 + 2



(b) 2 + 3



(c) 1 + 2 + 3

Рис. 3: Суммы Минковского многоугольников

Задача Н. Делоне

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

Этой ночью на «Адмирал Бенбоу» нападет шайка пиратов, желающая завладеть картой сокровищ. Но юный Джимми Хокинс уже готов их встретить. Его знакомый пират Делоне сконструировал хитрый механизм, представляющий собой секретные ловушки, а сам Джим сидит на втором этаже с сетями.

Ловушки представляют собой набор скрытых рычагов, расположенных на плоскости. Некоторые рычаги объединены между собой в ловушки. Ловушка может быть образована любыми тремя рычагами при условии, что ее площадь не равна нулю. При этом общая площадь любой пары ловушек равна нулю. Также гарантируется, что суммарная площадь, покрытая ловушками, максимальна.

Система работает следующим образом. Как только пират оказывается на территории одной из ловушек, Джим кидает сверху круглую сеть, в которую оказывается пойман пират. При этом край сети, имеющий форму окружности, касается всех трех рычагов, образующих ловушку, в которую попал пират.

Единственный недостаток этой системы заключается в том, что если в момент бросания сети она заденет своей внутренней точкой какой-нибудь рычаг, не имеющий отношения к нужной ловушке, вся система сломается. Джим хочет проверить, является ли система надежной. Более точно, он хочет выяснить, существует ли такая ловушка, что внутри сети, относящейся к этой ловушке (окружности, описанной вокруг треугольника, образованного соответствующими рычагами), содержится еще хотя бы один рычаг.

Формат входных данных

В первой строке входного файла задано число t ($1 \leq t \leq 5$) — число тестов. Далее идут описания t тестов.

В первой строке описания теста задано число n ($3 \leq n \leq 50\,000$) — число рычагов в сичтеме и число m — число ловушек.

Далее в n строках заданы целые числа x_i, y_i ($|x_i|, |y_i| \leq 10\,000$) — координаты рычагов.

Далее в m строках заданы по три числа — описания ловушек.

Каждое описание состоит из трёх чисел a_j, b_j, c_j ($1 \leq a_j, b_j, c_j \leq n$) — номера рычагов, из которых состоит i -я ловушка.

Формат выходных данных

Для каждого теста в отдельной строке выведите «YES», если система является надёжной, то есть, в любой описанной вокруг рычагов ловушки окружности больше не содержится рычагов или «NO» в ином случае.

Пример

стандартный ввод	стандартный вывод
2	YES
4 2	NO
0 1	
0 -1	
-2 0	
2 0	
1 2 3	
1 2 4	
4 2	
0 1	
0 -1	
-2 0	
2 0	
3 4 2	
3 4 1	

Задача I. Месть дракона

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

Один сказочный король до смерти ненавидел драконов. Он собрал огромную армию и убил их всех до единого. Но он упустил из вида одно небольшое драконье яйцо, приняв его за камень, так что из этого яйца вылупился последний дракон. Быстро установив виновника геноцида остальных драконов, последний дракон сильно разозлился и разрушил всё королевство. Армии, чтобы от него отбиваться, совсем не осталось, поскольку она вся была потрачена на уничтожение всех остальных драконов.

Сказочное королевство представляет собой координатную плоскость. Король с остатками армии укрылся в каменном дворце, который находится в точке с координатами (x_p, y_p) . Дворец надёжно защищён, и дракон не способен его разрушить. Поняв это, дракон спрятался в точке с координатами (x_d, y_d) в надежде подкараулить короля, когда тот покинет дворец. Король вскоре узнал о местонахождении дракона и теперь пытается оценить опасность ситуации.

Если король будет находиться за пределами дворца, то в любой момент дракон может это заметить и вылететь за ним из своего логова. Вылет дракона немедленно будет замечен стражей, и король тут же начнёт равномерно прямолинейно двигаться в сторону дворца, пока не окажется в безопасности за его стенами. Дракон же жаждет мести, поэтому просто в каждый момент времени будет равномерно прямолинейно двигаться точно в направлении текущего местоположения короля. Король уже стар, а дракон в самом расцвете сил, так что скорость перемещения короля ровно в два раза меньше, чем скорость перемещения дракона. Король хотел бы определить, насколько сильно дракон ущемляет свободу его передвижения, поэтому желает узнать площадь безопасного участка плоскости, по которому он может разгуливать без опасений, что в случае вылета дракон догонит его раньше, чем он доберётся до дворца.

Формат входных данных

Первая строка содержит два целых числа через пробел: x_p и y_p ($-1000 \leq x_p, y_p \leq 1000$) — координаты дворца.

Вторая строка содержит два целых числа через пробел: x_d и y_d ($-1000 \leq x_d, y_d \leq 1000$) — координаты логова дракона.

Эти две точки не совпадают.

Формат выходных данных

Выведите единственное вещественное число — площадь безопасной части плоскости. Абсолютная или относительная погрешность не должна превышать 10^{-6} .

Пример

стандартный ввод	стандартный вывод
0 0	0.916297857297023
1 0	

Задача J. Пересечение полуплоскостей

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

Эта задача не имеет условия.

Формат входных данных

Первая строка входного файла содержит единственное число n — число полуплоскостей, площадь пересечения которых вам требуется посчитать ($3 \leq n \leq 2000$). Следующие n строк содержат по три целых числа a_i, b_i, c_i , по модулю не превосходящих 10000, задающие полуплоскость $((x, y)$ принадлежит i -й полуплоскости, если $a \cdot x + b \cdot y + c \geq 0$). Гарантируется, что площадь пересечения конечна.

Формат выходных данных

Одно вещественное число — площадь пересечения полуплоскостей с точностью до 10^{-6} .

Пример

стандартный ввод	стандартный вывод
3 1 0 0 0 1 0 -1 -1 1	0.5000000000

Задача К. Целочисленные точки

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

Многоугольник (не обязательно выпуклый) на плоскости задан координатами своих вершин. Требуется подсчитать количество точек с целочисленными координатами, лежащих внутри него (но не на его границе).

Формат входных данных

В первой строке вводится n ($1 \leq n \leq 1000$) — число вершин многоугольника. В последующих n строках идут координаты (x_i, y_i) вершин многоугольника в порядке обхода по часовой стрелке. x_i и y_i — целые числа, по модулю не превосходящие 10^6 .

Формат выходных данных

Выведите одно число — количество искомых точек.

Пример

стандартный ввод	стандартный вывод
4 1 1 1 2 2 2 2 1	0

Задача L. Растягивание плоскости

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	5 секунд
Ограничение по памяти:	1024 мегабайта

Игорь очень любит геометрию, а поэтому он купил себе плоскость, на которой отмечены n точек, i -я из них имеет координаты (x_i, y_i) .

Посмотрев на эти точки, Игорь быстро нашёл пару самых удалённых. Однако этого ему было мало, а поэтому для q чисел $\alpha_1, \alpha_2, \alpha_3, \dots, \alpha_q$ Игорь хочет узнать, каким станет максимальное расстояние между парой точек, если растянуть плоскость в α_j раз по x -координате.

Более формально, у Игоря есть q запросов, в j -м из которых для числа α_j Игорь хочет найти расстояние между двумя наиболее удалёнными точками в множестве, состоящем из n точек с координатами $(x_i \cdot \alpha_j, y_i)$. Помогите Игорю ответить на эти запросы.

Формат входных данных

Каждый тест состоит из нескольких наборов входных данных. В первой строке вводятся два целых числа t и g ($1 \leq t \leq 250\,000$, $0 \leq g \leq 9$) — число наборов входных данных и номер группы тестов, под дополнительные ограничения которой подходит данный тест. Далее следуют описания наборов входных данных.

В первой строке каждого набора входных данных вводятся два целых числа n и q ($2 \leq n \leq 500\,000$, $1 \leq q \leq 500\,000$) — количество точек и количество запросов.

В следующих n строках вводятся описание точек, в каждой строке вводятся по два целых числа x_i и y_i ($-10^9 \leq x_i, y_i \leq 10^9$) — координаты i -й точки. Гарантируется, что координаты всех точек в каждом наборе входных данных различны.

В следующих q строках вводятся описания запросов, в каждой строке вводится по одному **вещественному** числу α_j ($1 \leq \alpha_j \leq 10^9$) — коэффициенты, на которые будут умножаться x -координаты точек в j -м запросе.

Обозначим за N сумму n_i по всем наборам входных данных, а за Q — сумму q_i по всем наборам входных данных. Гарантируется, что $N, Q \leq 500\,000$.

Формат выходных данных

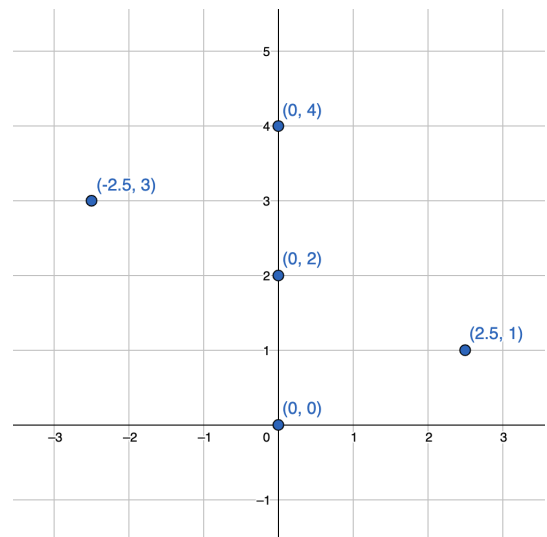
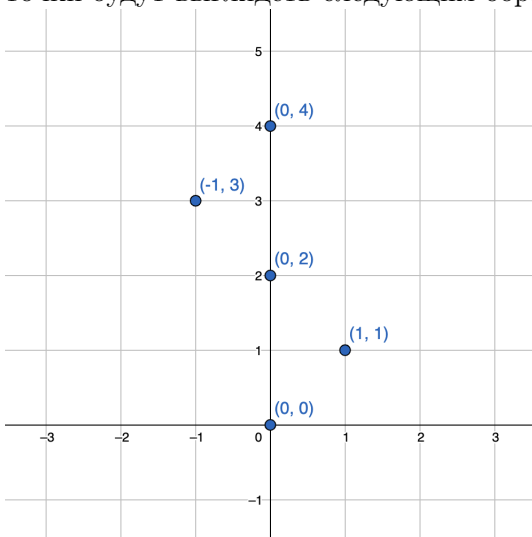
Для каждого набора входных данных выведите q строк, в i -й строке должно содержаться единственное вещественное число — ответ на i -й запрос. Ответ будет считаться правильным, если его абсолютная или относительная погрешность не превышает 10^{-6} . Более формально, если a — ваш ответ, а b — ответ жюри, то должно выполняться $\frac{|a-b|}{\max(b,1)} \leq 10^{-6}$.

Пример

стандартный ввод	стандартный вывод
2 0	4.000000
5 2	5.385165
0 0	28.000000
1 1	15.000000
0 2	17.500000
-1 3	21.000000
0 4	
1	
2.5	
8 4	
0 0	
6 11	
7 13	
4 14	
0 15	
-4 14	
-7 13	
-6 11	
2	
1	
1.25	
1.5	

Замечание

В первом наборе входных данных при растяжении с коэффициентом 1 и с коэффициентом 2.5 точки будут выглядеть следующим образом:



При растяжении с коэффициентом 1 наиболее удалёнными точками будут точки с номерами 1 и 5, их координаты будут равны $(0, 0)$ и $(0, 4)$.

При растяжении с коэффициентом 2.5 наиболее удалёнными точками будут точки с номерами 2 и 4, их координаты будут равны $(2.5, 1)$ и $(-2.5, 3)$.

Во втором наборе входных данных максимальное расстояние будет достигаться следующими парами точек:

- в первом запросе максимальное расстояние будет достигаться между точками с номерами 3 и 7, их координаты будут равны $(14, 13)$ и $(-14, 13)$,

- во втором запросе максимальное расстояние будет достигаться между точками с номерами 1 и 5, их координаты будут равны $(0, 0)$ и $(0, 15)$,
- в третьем запросе максимальное расстояние будет достигаться между точками с номерами 3 и 7, их координаты будут равны $(8.75, 13)$ и $(-8, 75, 13)$,
- в четвёртом запросе максимальное расстояние будет достигаться между точками с номерами 3 и 7, их координаты будут равны $(10.5, 13)$ и $(-10.5, 13)$.

Система оценки

Задача М. Парк

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

В столице Байтленда есть огороженный парк, территория которого представляет собой прямоугольник. Деревья и посетители в парке представлены в виде кругов.

В парке четыре входа, по одному в каждом углу (1 — внизу слева, 2 — внизу справа, 3 — вверху справа, 4 — вверху слева). Посетители могут входить и выходить из парка только через входы. Посетители могут входить и выходить из парка, когда они касаются обеих сторон угла соответствующего входа. Посетители могут свободно перемещаться по парку, но они не могут перекрывать деревья и заборы.

Ваша задача — рассчитать для каждого посетителя, учитывая вход, в который они войдут в парк, через какие выходы они могут выйти из парка.

Формат входных данных

Первая строка ввода содержит два целых числа n и m ($1 \leq n \leq 2000$ $1 \leq m \leq 100\,000$) — количество деревьев в парке и количество посетителей.

Вторая строка ввода содержит два целых числа w и h ($1 \leq w, h \leq 10^9$) — ширину и высоту парковой зоны. Нижний левый угол имеет координаты $(0, 0)$, а верхний правый угол имеет координаты (w, h) .

После этого есть n строк, которые описывают деревья. Каждая строка содержит три целых числа x , y и r , что означает, что центр i -го дерева (x, y) и его радиус r . Деревья не перекрывают друг друга или забор.

Наконец, есть m строк, которые описывают посетителей. Каждая строка содержит два целых числа r и e — радиус посетителя и вход, в который они войдут в парк.

Кроме того, ни одно дерево не перекрывает квадратную область $2k \times 2k$ в каждом углу, где k — радиус наибольшего посетителя.

Формат выходных данных

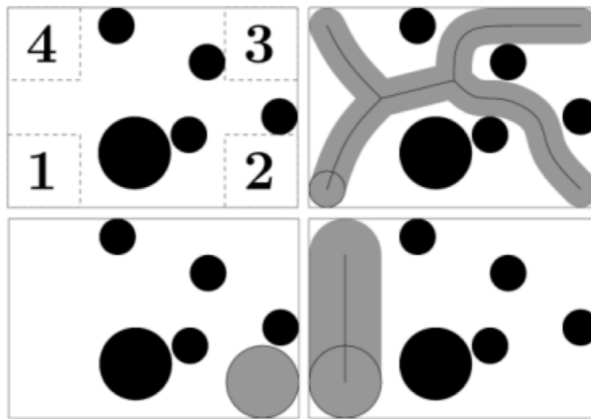
Вы должны вывести для каждого посетителя одну строку, содержащую входы, через которые они могут выйти из парка, в отсортированном порядке без пробелов между ними.

Пример

стандартный ввод	стандартный вывод
5 3	1234
16 11	2
11 8 1	14
6 10 1	
7 3 2	
10 4 1	
15 5 1	
1 1	
2 2	
2 1	

Замечание

На следующем рисунке показаны зоны входа и возможные маршруты для каждого посетителя:



Задача N. Большой треугольник

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 8 секунд
Ограничение по памяти: 1024 мегабайта

Есть странная особенность: если соединить между собой города Ростов, Таганрог, Шахты, то получится треугольник

«Невероятно, но факт»

В ЛКШ приезжают ученики из самых разных уголков России и зарубежья. Вы отметили на карте все города, откуда приехали участники ЛКШ.

Затем, вы решили подготовить интересную инфографику на основе этой карты. Первое что вы захотели сделать — это найти три города на этой карте, так чтобы получился треугольник площади ровно S .

Формат входных данных

В первой строке входных данных находится два целых числа n и S ($3 \leq n \leq 3333$, $1 \leq S \leq 2 \cdot 10^{18}$) — количество городов на карте и требуемая площадь треугольника.

В следующую n строках находится описание городов, по одной на строке. Каждый город описывается своими координатами x_i, y_i ($-10^9 \leq x_i, y_i \leq 10^9$).

Гарантируется, что все города находятся в различных точках. Также гарантируется, что никакие три города не лежат на одной прямой.

Формат выходных данных

Если решения не существует — выведите «No» (без кавычек).

Иначе выведите «Yes» (без кавычек), после чего выведите три пары координат — координаты городов, образующих треугольник площади S .

Примеры

стандартный ввод	стандартный вывод
3 7 0 0 3 0 0 4	No
4 3 0 0 2 0 1 2 1 3	Yes 0 0 1 3 2 0

Задача О. Фонтан

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	0.25 секунд
Ограничение по памяти:	256 мегабайт

Администрация одного института решила построить в холле фонтан. По плану администрации, фонтан должен иметь форму круга с максимально возможным радиусом. Дизайнеру сообщили, что холл института имеет вид прямоугольника, размером $X \times Y$ метров. Однако когда дизайнер стал выбирать место для фонтана, он столкнулся с серьезной проблемой: в холле института обнаружилось N круглых колонн, снести которые не представляется возможным.

Таким образом, у него появилась проблема: где следует поместить фонтан, чтобы он имел максимально возможный радиус и не имел ненулевого по площади пересечения с колоннами. Вам предстоит помочь ему в решении этой нелегкой задачи.

Формат входных данных

В первой строке входных данных содержатся вещественные числа X и Y , ($1 \leq X, Y \leq 10^4$). Будем считать, что прямоугольник холла расположен на координатной сетке так, что его углы имеют координаты $(0, 0)$, $(X, 0)$, (X, Y) и $(0, Y)$.

Во второй строке задается число N ($0 \leq N \leq 10$) — количество колонн.

Следующие N строк содержат параметры колонн — i -я строка содержит три вещественных числа X_i , Y_i и R_i — координаты центра и радиус i -й колонны ($R_i \leq X_i \leq X - R_i$, $R_i \leq Y_i \leq Y - R_i$, $0.1 \leq R_i \leq \min(X/2, Y/2)$; для любых $i \neq j$ $\text{sqrt}((X_i - X_j)^2 + (Y_i - Y_j)^2) \geq R_i + R_j$). Все вводимые числа разделены пробелами.

Формат выходных данных

Выведите максимальный возможный радиус фонтана.

Примеры

стандартный ввод	стандартный вывод
10 10 0	5.00000000
1 1000 0	0.50000000
10 10 4 1 1 1 9 9 1 1 9 1 9 1 1	4.65690613

Задача Р. Жопа

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	6 секунд
Ограничение по памяти:	256 мегабайт

Бывший председатель уважаемой олимпиады решил отдохнуть и сходить в баню. Баня является специальным помещением, где при высокой температуре воздуха люди моются и парятся. Как правило, при посещении бани люди сидят на деревянной лавочке. Такая лавочка и находилась в бане, которую посещает председатель.

Сама лавочка имеет деревянную прямоугольную сидушку размером $x \times y$. Именно на эту сидушку опустится жопа председателя, когда тот пойдет в баню. Но есть один нюанс. Сама сидушка закреплена металлическими гвоздями, и как известно, металл довольно сильно греется при высокой температуре. По этой причине, если жопа будет касаться гвоздей, то председателю будет очень неприятно и все положительные ощущения от посещения бани пройдут. Всего на сидушке закреплено n гвоздей, i -й из гвоздей находится в точке с координатами (x_i, y_i) .

Жопа председателя состоит из двух полужолий, представляющих из себя две касающиеся окружности одинакового радиуса. Помогите председателю узнать, какой максимальный радиус может иметь его жопа, чтобы под ней не было ни одного гвоздя и она полностью поместилась на лавочку (то есть никакой своей частью не выходит за пределы сидушки). **Обратите внимание, что гвоздь может находиться в центре жопы (в точке соприкосновения двух полужолий).**

Формат входных данных

В первой строке дано единственное целое число t ($1 \leq t \leq 1000$) — количество наборов входных данных. В следующих строках содержатся описания каждого набора входных данных.

В первой строке каждого описания даны два целых числа x и y ($2 \leq x, y \leq 10\,000$) — размеры сидушки.

Во второй строке вводится одно целое число n ($1 \leq n \leq 1000$) — количество гвоздей.

В следующих n строках вводятся по два целых числа x_i, y_i ($1 \leq x_i \leq x - 1, 1 \leq y_i \leq y - 1$) — координаты гвоздей.

Гарантируется, что координаты всех гвоздей различны в каждом из наборов входных данных.

Гарантируется, что сумма n по всем наборам входных данных не превосходит 1000.

Формат выходных данных

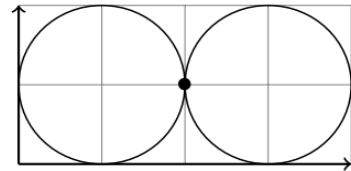
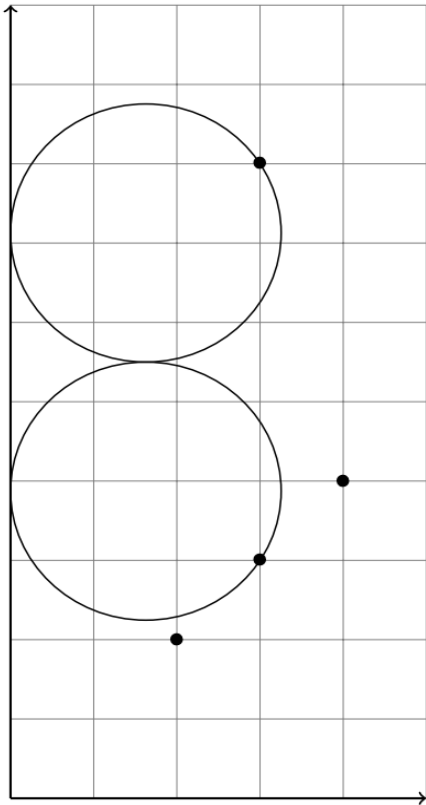
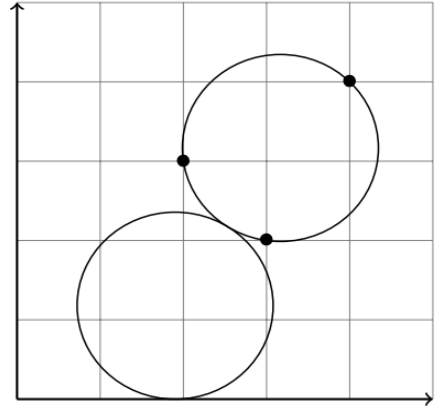
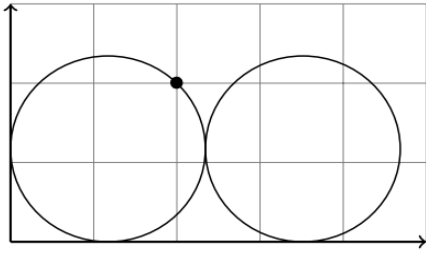
Для каждого набора входных данных выведите вещественное число r — максимальный радиус полужолия с абсолютной или относительной погрешностью не более 10^{-6} .

Примеры

стандартный ввод	стандартный вывод
4 5 3 1 2 2 5 5 3 3 2 2 3 4 4 5 10 4 3 8 2 2 3 3 4 4 4 2 1 2 1	1.1715729 1.1785113 1.6270167 1.0000000
1 10 10 10 1 4 2 7 3 2 5 8 6 1 6 2 6 7 7 2 7 8 8 5	1.9649222
1 1505 1671 10 38 912 147 740 149 266 259 459 360 802 513 523 692 299 717 1185 1114 362 1363 1536	368.1802590

Замечание

Расположения жоп, соответствующих ответам на первый тест из условия.



Задача Q. Дамба

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 3 секунды
Ограничение по памяти: 512 мегабайт

Есть изначально пустая дамба. Дамба может вместить не больше L литров воды.

Каждое утро в дамбу будет добавляться v_i литров воды температурой t_i градусов. Перед тем, как вода нальётся, можно вылить любое количество воды, которая сейчас находится в дамбе.

Для каждого дня требуется посчитать максимальную температуру воды в дамбе, если на момент этого дня в ней должно быть ровно L литров воды, а так же в любой момент времени дамба не должна переполниться.

Обратите внимание, что ответы не зависят друг от друга, то есть для того, чтобы максимизировать температуру в 3-й день не требуется, чтобы в 1-й первый или во 2-й день дамба содержала ровно L литров воды, но требуется, чтобы в 3-й день в ней было ровно L литров.

Если в дамбе находилось v_1 литров воды с температурой t_1 и в неё добавилось v_2 литров воды температурой t_2 , то итоговая температура будет равна $\frac{t_1 \cdot v_1 + t_2 \cdot v_2}{v_1 + v_2}$.

Формат входных данных

Первая строка входных данных содержит два числа n и L ($1 \leq n \leq 5 \cdot 10^5, 1 \leq L \leq 10^9$) — количество дней и вместимость дамбы.

Далее следуют n строк. i -я строка содержит два числа t_i и v_i ($0 \leq t_i \leq 10^9, 1 \leq v_i \leq L$) — температура и объём добавленной воды в i -й день.

Гарантируется, что $v_1 = L$.

Формат выходных данных

Выведите n чисел. i -е число должно быть равно максимальной температуре в i -й день. Погрешность ответа не должна превосходить 10^{-6} .

Примеры

стандартный ввод	стандартный вывод
3 10 10 10 20 5 4 3	10 15.0000000000 13.2000000000
4 15 0 15 2 5 3 6 4 4	0 0.6666666667 1.8666666667 2.9333333333
4 15 1000000000 15 9 5 8 6 7 4	1000000000 666666669.6666666667 400000005.0000000000 293333338.8666666667

Задача R. Караваны

Имя входного файла: `caravans.in`
Имя выходного файла: `caravans.out`
Ограничение по времени: 4 секунды
Ограничение по памяти: 256 мегабайт

В этой задаче вам нужно грабить караваны.

В пустыне есть n оазисов (пусть они находятся в точках на плоскости). Иногда караваны отправляются от одного оазиса к другому оазису. Чтобы грабить указатели, нужно уметь предсказывать их пути. Но как это сделать? Ответ знает Номад. Скорость караванов постоянна, и они пытаются минимизировать максимальное время вне оазисов. Поэтому можно считать путь караванов ломаной. Вам известны несколько пар оазисов, и вам нужно найти максимальную длину отрезка вне оазисов оптимального пути каравана, который идет от первого оазиса ко второму. Все оазисы находятся в разных точках и никакие три оазиса не лежат на одной прямой.

Формат входных данных

В первой строке дано одно целое число n — количество оазисов ($3 \leq n \leq 100\,000$).

В следующих n строках даны пары целых чисел x_i, y_i — координаты оазисов ($0 \leq x_i, y_i \leq 10\,000$).

В следующей строке дано одно целое число q — количество караванов ($1 \leq q \leq 100\,000$).

В следующих q строках даны пары целых чисел s_i, t_i — стартовый и конечный оазис на пути каравана ($1 \leq s_i, t_i \leq n$).

Формат выходных данных

Выведите q чисел — длины искомых максимальных отрезков на пути, с относительной или абсолютной погрешностью 10^{-9} .

Пример

<code>caravans.in</code>	<code>caravans.out</code>
3	50.990195135928
0 0	100.498756211209
50 10	100.498756211209
150 0	
3	
1 2	
1 3	
2 3	

Задача S. Футбол 2

Имя входного файла: `football2.in`
Имя выходного файла: `football2.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

256 мегабайт

На футбольном поле размером $x \times y$ находятся n футболистов. Они уже очень устали и стоят на месте, но ждут, куда упадет мяч, чтобы побежать к нему. Футболист бежит к мячу в том случае, если мяч упал к этому футболисту ближе, чем к любому другому футболисту. Требуется определить для каждого футболиста границы зоны, при попадании в которую он побежит к мячу, если известно, что она представляет собой многоугольник.

Формат входных данных

В первой строке входного файла заданы три целых числа x , y и n ($2 \leq x, y \leq 10^5$, $1 \leq n \leq 30\,000$). Следующие n строк содержат целые координаты футболистов $x_i y_i$ ($0 < x_i < x$, $0 < y_i < y$). Никакие два футболиста не стоят в одной точке.

Формат выходных данных

В выходной файл выведите n строк. В каждой из строк первое число — количество вершин зоны k_i , далее k_i чисел — координаты вершин $x_{ij} y_{ij}$ в порядке обхода против часовой стрелки, начиная с самой нижней из самых левых вершин зоны. Вещественные числа выводите с максимальной точностью.

Пример

football2.in	football2.out
4 4 4	4 0.00000000 0.00000000 2.00000000 0.00000000 2.00000000
1 1	4 0.00000000 2.00000000 2.00000000 2.00000000 2.00000000
1 3	4 2.00000000 0.00000000 4.00000000 0.00000000 4.00000000
3 1	4 2.00000000 2.00000000 4.00000000 2.00000000 4.00000000
3 3	