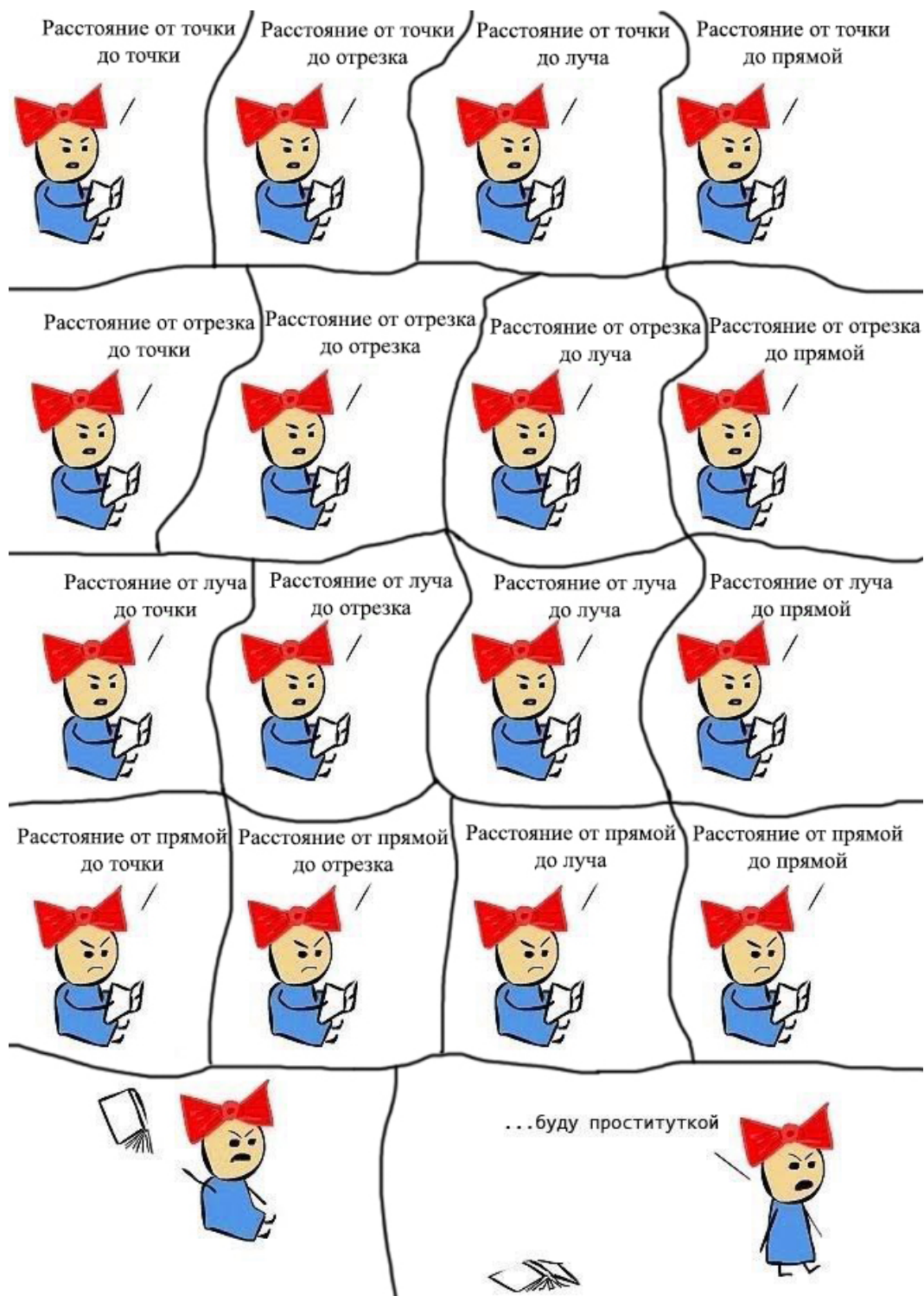


Задача А. 16

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

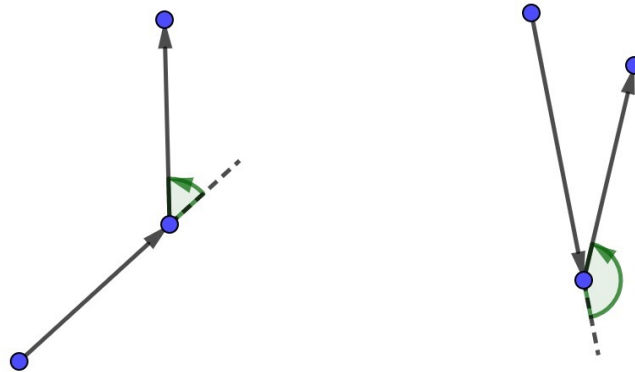
Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

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 .

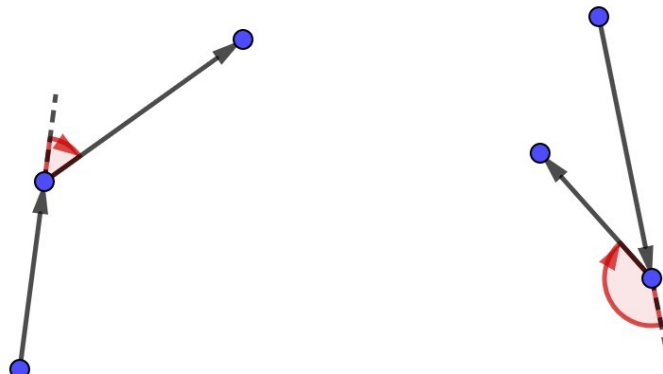
Сделав так, он нарисует ориентированную ломаную на этих вершинах, проведя направленные отрезки из каждой точки в следующую в выбранном порядке точку. То есть для всех $1 \leq i \leq n-1$ он проведет направленный отрезок из точки A_{p_i} в точку $A_{p_{i+1}}$. Он хочет, чтобы получившаяся ломаная удовлетворяла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

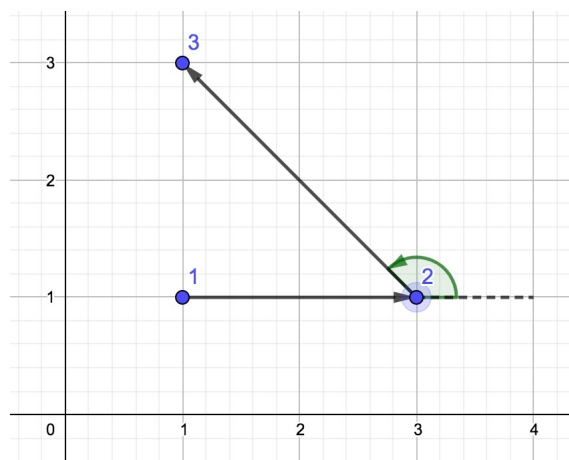
Если подходящей перестановки не существует выведите -1 . Иначе выведите n чисел $p_1, p_2, \dots, p_n$ — найденную перестановку ($1 \leq p_i \leq n$ и все $p_1, p_2, \dots, p_n$ различны). Если подходящих перестановок несколько, выведите любую.

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

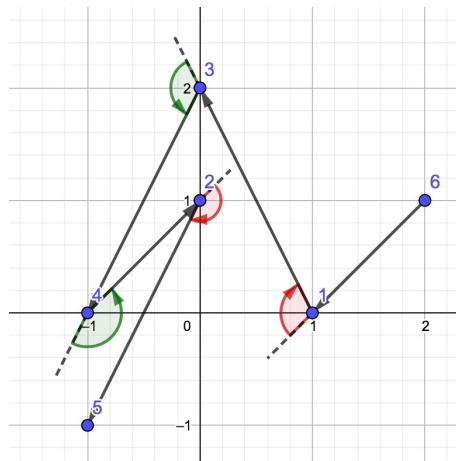
Замечание

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

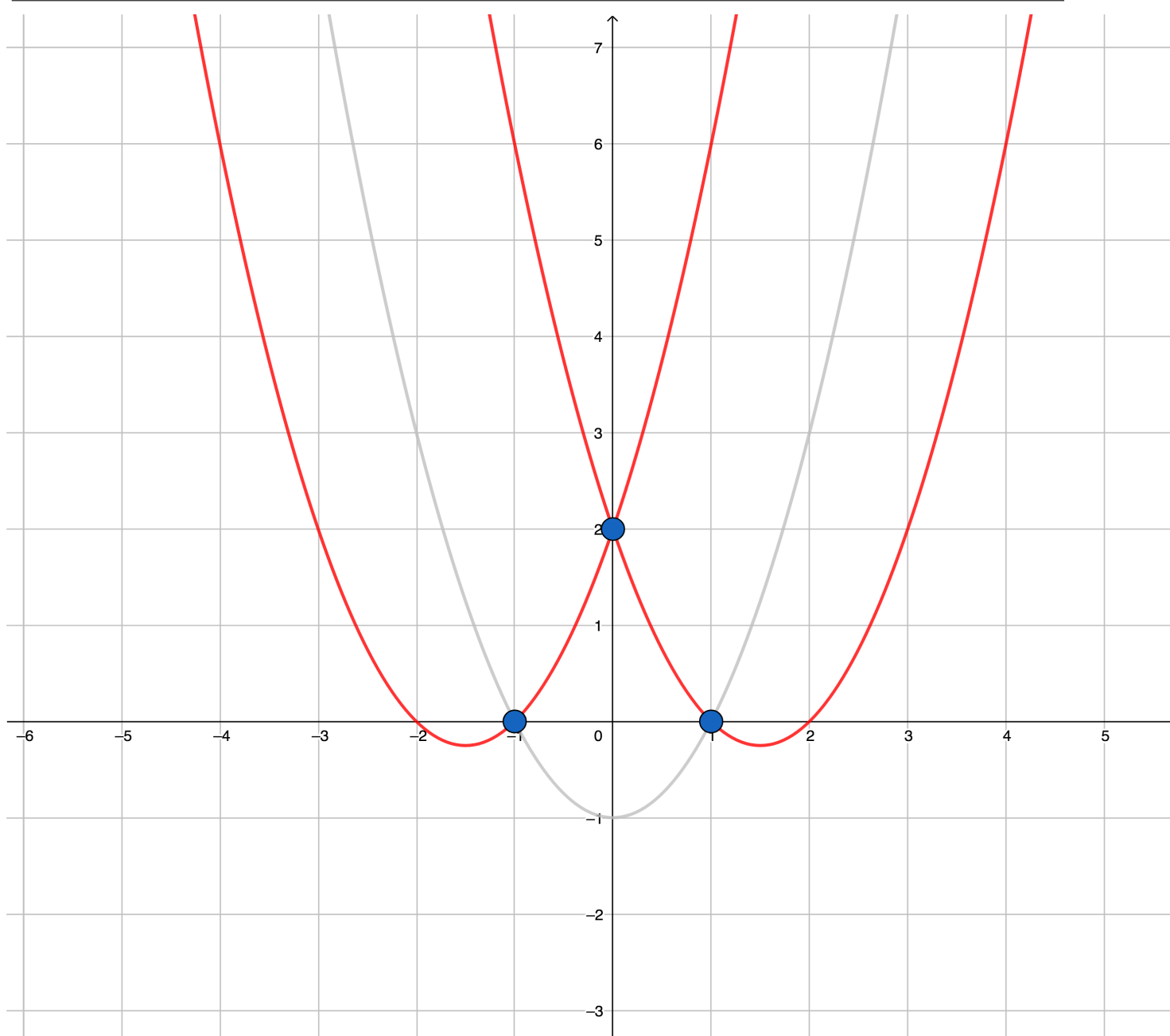
В единственной строке выведите единственное число — количество U -образных парабол, проходящих хотя бы через 2 точки и не содержащих никаких других точек в своей внутренней области (не считая границы).

Примеры

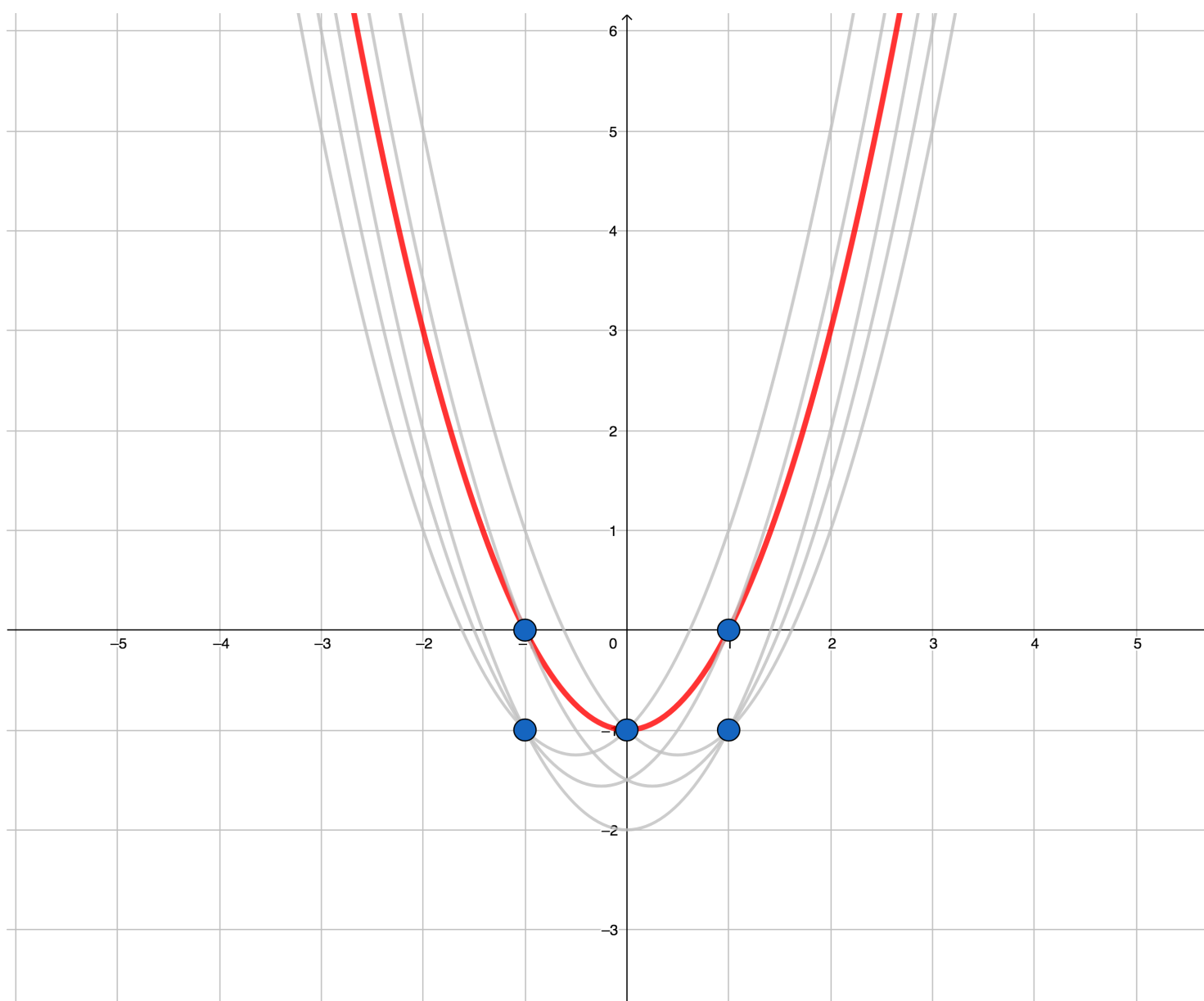
стандартный ввод	стандартный вывод
3 -1 0 0 2 1 0	2
5 1 0 1 -1 0 -1 -1 0 -1 -1	1

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. Бульдозер

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2.5 секунд
Ограничение по памяти:	256 мегабайт

Королевство JOI известно добычей золота. Каждый год они используют бульдозер для его добычи.

Земля королевства описывается плоскостью с декартовыми координатами. Есть n мест, i -е место имеет координаты (x_i, y_i) . В каждом месте возможно добыть золото или камни, но нельзя одновременно и то, и другое.

Если в i -м месте можно добыть золото, то его ценность там равна v_i . Если же там камни, то, чтобы избавиться от них, необходимо потратить c_i .

Мы используем бульдозер следующим образом. Сначала мы выбираем две параллельные прямые, затем добываем всё золото и камни, находящиеся между этими прямыми, включая их границы.

Выгода равна сумме ценностей золота за вычетом трат на избавление от камней. Необходимо максимизировать выгоду.

Формат входных данных

Первая строка содержит целое число n ($1 \leq n \leq 2000$) — количество мест.

Далее следуют n строк, i -я из которых содержит три целых числа x_i, y_i, w_i ($|x_i|, |y_i|, |w_i| \leq 10^9$, $w_i \neq 0$) — координаты места (x_i, y_i) и специальное число.

- Если $w_i \geq 1$, то i -е место содержит золото, и $v_i = w_i$.
- Если $w_i \leq -1$, то i -е место содержит камни, и $c_i = -w_i$.

Гарантируется, что все точки различны.

Формат выходных данных

Выведите одно число — ответ.

Примеры

стандартный ввод	стандартный вывод
5 -5 5 -2 2 5 10 1 4 -2 4 -5 4 -2 2 7	19
6 0 0 6 1 0 -2 2 0 8 0 1 -2 1 1 5 2 1 -2	15
5 0 0 2 4 0 2 3 2 -1 1 2 2 1 1 -1	5
2 0 0 -1 1 0 -1	0
15 10 3 30 5 10 -17 4 -5 14 0 -3 -9 -2 3 17 6 9 -19 -9 -6 -14 -2 -3 10 -3 -3 30 8 1 -28 9 -9 -5 7 -5 -24 -8 -10 5 -7 2 20 10 -3 -13	107

Задача F. 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 дешевую 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 Г. 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 $i \neq j$ $\sqrt{(X_i - X_j)^2 + (Y_i - Y_j)^2} \geq R_i + R_j$). 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 Н. Тигры HARD

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

В заповеднике живут q тигров. Чтобы следить за положением тигров на территории заповедника, используются ошейники с радиомаяком. Ошейник у каждого тигра имеет радиомаяк с уникальным сигналом. Система обнаружения настраивается на приём сигнала радиомаяка от i -го тигра последовательно для i от 1 до q .

Для приёма сигнала на территории заповедника установлено n приёмников в точках с координатами $(x_1, y_1), \dots, (x_n, y_n)$. Система обнаружения позволяет сотруднику заповедника за один запрос выбрать любые m ($3 \leq m \leq n$) приёмников. Выбранные приёмники должны являться вершинами выпуклого многоугольника. Система определяет, находится ли радиомаяк i -го тигра внутри этого многоугольника.

Сотрудник заповедника должен локализовать положение каждого тигра. Положение i -го тигра считается локализованным, если удалось определить такое множество приёмников, являющихся вершинами выпуклого многоугольника, что внутри этого многоугольника находится тигр, но нет других приёмников.

Для того, чтобы локализовать положение каждого из тигров, сотруднику разрешается сделать не более k запросов.

После того как положение i -го тигра локализовано, система автоматически переходит к приёму сигналов от следующего тигра, пока положение всех q тигров не будет локализовано.

Гарантируется, что никакие три приёмника не лежат на одной прямой, и ни один тигр не находится на прямой, проходящей через два приёмника. Гарантируется, что существует хотя бы один выпуклый многоугольник с вершинами в приёмниках, внутри которого находится тигр.

Требуется написать программу, которая взаимодействует с программой жюри и локализует положение каждого тигра.

Формат входных данных

Это интерактивная задача.

Сначала на вход подаётся информация об установленных в заповеднике приёмниках и количестве тигров.

Первая строка входных данных содержит целое число n — количество приёмников ($3 \leq n \leq 5000$). Последующие n строк описывают координаты приёмников, j -я из этих строк содержит два целых числа x_j и y_j — координаты j -го приёмника ($-10^9 \leq x_j, y_j \leq 10^9$). Следующая строка содержит число целое число q — количество тигров ($1 \leq q \leq 2000$).

Для локализации положения тигров необходимо выполнять запросы к системе обнаружения, роль которой выполняет программа жюри.

Для каждого теста зафиксировано число k — максимальное количество запросов к системе обнаружения для локализации положения одного тигра. Гарантируется, что k запросов достаточно, чтобы решить задачу для соответствующих данных. Это число не сообщается программе-решению. Если программа-решение делает более k запросов для определения местоположения одного из тигров, на этом тесте она получает в качестве результата тестирования «Неверный ответ». При $n > 2500$ $k = 10000$. При $n \leq 2500$ $k = 60$.

Запрос к системе обнаружения начинается с символа «?», за которым следует целое число m — количество выбранных в запросе приёмников ($3 \leq m \leq n$), и m различных целых чисел p_i — номера приёмников, перечисленные в порядке обхода многоугольника по или против часовой стрелки ($1 \leq p_i \leq n$).

В ответ программа получает строку «Yes», если тигр находится внутри многоугольника, образованного приёмниками с номерами $p_1, \dots, p_m$, и строку «No» в противном случае.

После того, как положение тигра локализовано, программа-решение должна вывести строку, начинающуюся с символа «!», за которым следует целое число m — количество выбранных приёмников ($3 \leq m \leq n$), и m различных целых чисел p_i — номера приёмников, перечисленные в порядке

обхода многоугольника по или против часовой стрелки ($1 \leq p_i \leq n$). Эта строка означает, что внутри выпуклого многоугольника, образованного приёмниками с номерами $p_1, \dots, p_m$, находится тигр и нет других приёмников.

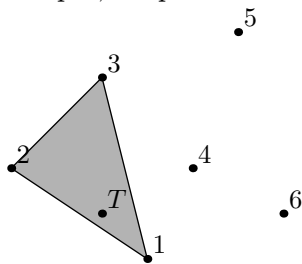
Ответное сообщение от программы жюри отсутствует, и программа-решение должна немедленно приступить к поиску следующего тигра. Локализовав положение тигра с номером q , программа-решение должна завершить работу.

Тигры не перемещаются во время работы системы обнаружения. Координаты тигров в каждом тесте фиксированы и не меняются в процессе тестирования.

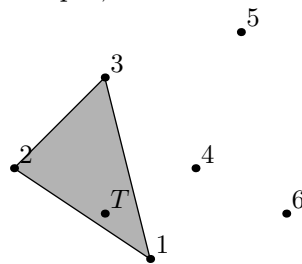
Если существует несколько правильных многоугольников, локализующих положение тигра, можно вывести любой из них.

На рисунке продемонстрирована процедура локализации положения каждого из тигров из приведенного ниже примера.

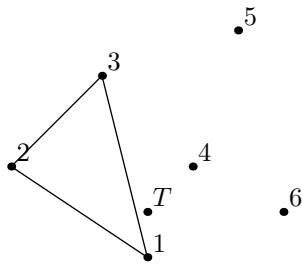
Тигр 1, запрос 1:



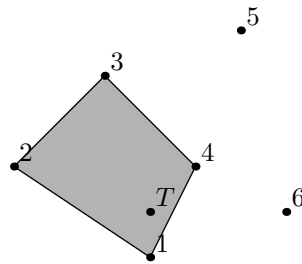
Тигр 1, ответ:



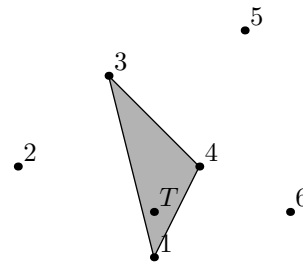
Тигр 2, запрос 1:



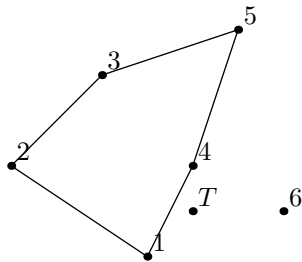
Тигр 2, запрос 2:



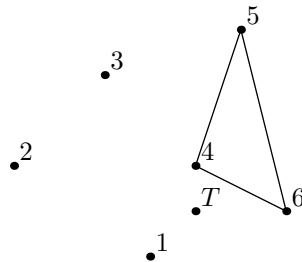
Тигр 2, ответ:



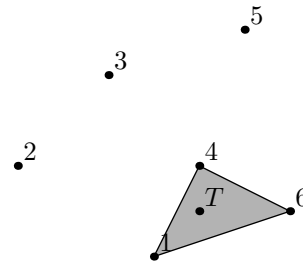
Тигр 3, запрос 1:



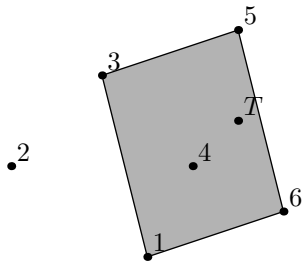
Тигр 3, запрос 2:



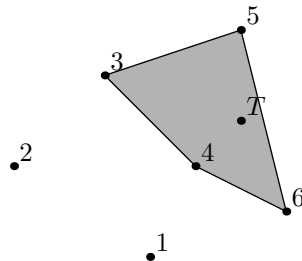
Тигр 3, ответ:



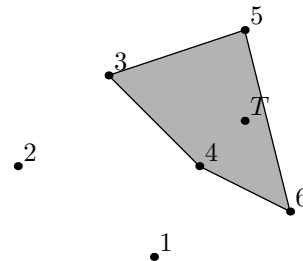
Тигр 4, запрос 1:



Тигр 4, запрос 2:



Тигр 4, ответ:



Пример

стандартный ввод	стандартный вывод
6	
3 0	
0 2	
2 4	
4 2	
5 5	
6 1	
4	
Yes	? 3 1 2 3
	! 3 1 2 3
	? 3 1 2 3
No	
	? 4 1 2 3 4
Yes	
	! 3 4 3 1
	? 5 2 3 5 4 1
No	
	? 3 4 5 6
No	
	! 3 1 4 6
	? 4 1 3 5 6
Yes	
	? 4 4 3 5 6
Yes	
	! 4 4 3 5 6

Замечание

Приведённые примеры иллюстрируют взаимодействие программы-решения с программой жюри «по шагам», для чего в них добавлены дополнительные пустые строки. При реальном тестировании лишние пустые строки вводиться не будут, выводить пустые строки также не требуется.

Задача I. 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

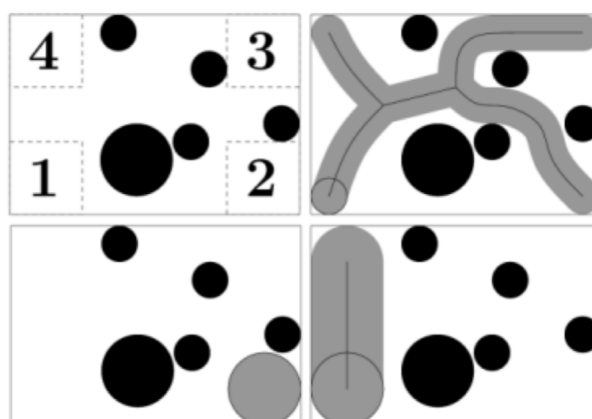
Вы должны вывести для каждого посетителя одну строку, содержащую входы, через которые они могут выйти из парка, в отсортированном порядке без пробелов между ними.

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 J. 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.

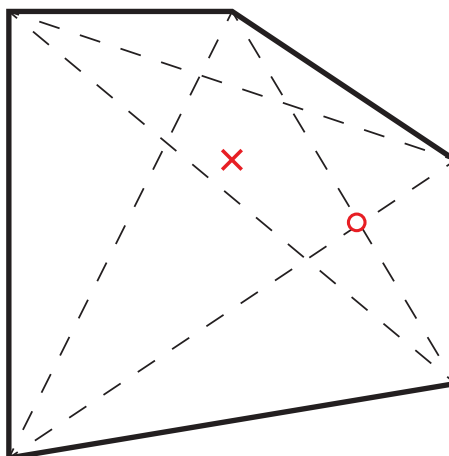
В i -й строке выходного файла выведите одно число — максимальное расстояние, на котором Вася может спрятаться от Пети, когда Петя будет сажать дерево в i -й день. Ваш ответ будет считаться правильным, если его абсолютная или относительная погрешность относительно правильного ответа составит не более 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 К. Формула 42

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	512 мегабайт

Для проведения гран-при Флатландии планируется построить новую трассу Формулы 42. Был приглашен известный проектировщик трасс Герман Килькин, и он сделал чертежи трассы. Однако после начала строительства выяснилось, что с проектом есть небольшая проблема.

Трасса Формулы 42 представляет собой территорию, ограниченную с двух сторон барьерами. Каждый барьер представляет собой замкнутую ломаную, которая является границей выпуклого многоугольника. Внутренний барьер находится полностью внутри внешнего и не имеет с ним общих точек. Болид Формулы 42 представляет собой круг. Радиус круга не регламентирован, и считается, что соревнования получаются тем зрелищнее, чем больше радиус болида.

Проблема в проекте трассы заключается в том, что Герман изобразил на чертеже отдельно форму внешнего барьера и отдельно — форму внутреннего. При этом их взаимное расположение не указано. Теперь строители трассы могут выбрать произвольное взаимное расположение барьеров. Они хотят расположить барьеры таким образом, чтобы гонка получилась максимально зрелищной. Каждый из барьеров можно перемещать параллельным переносом, но нельзя поворачивать, отражать или другим образом изменять.

Требуется выбрать такое взаимное расположение барьеров, чтобы получившуюся трассу можно было использовать для гонки с максимальным размером гоночного болида. Трассу можно использовать с болидом радиуса r , если болид такого радиуса, находясь на трассе, может совершить полный круг вокруг внутреннего барьера. В процессе движения никакая точка болида не должна оказываться строго внутри внутреннего барьера или строго снаружи от внешнего барьера. Касания барьеров разрешены. Формально: для точки P внутри замкнутого барьера существует такая непрерывная замкнутая кривая, что точка P лежит внутри этой кривой, и для любой точки этой кривой круг с радиусом r с центром в этой точке лежит внутри внешнего барьера и снаружи от внутреннего барьера, касания разрешаются.

Помогите разобраться с планом Германа и понять, для какого максимального радиуса гоночного болида возможно построить гоночную трассу в соответствии с имеющимися чертежами внешнего и внутреннего барьеров.

Формат входных данных

Входные данные содержат описание двух многоугольников, границы которых задают формы внешнего и внутреннего барьеров, соответственно.

В первой строке находится целое число n — количество вершин в первом многоугольнике ($3 \leq n \leq 100$). В следующих n строках даны по два целых числа x_i и y_i — координаты вершин первого многоугольника ($0 \leq x_i, y_i \leq 1000$).

В следующей строке находится целое число m — количество вершин во втором многоугольнике ($3 \leq m \leq 100$). В следующих m строках даны по два целых числа x'_i и y'_i — координаты вершин второго многоугольника ($0 \leq x'_i, y'_i \leq 1000$).

Вершины каждого многоугольника заданы в порядке обхода против часовой стрелки. Многоугольники являются выпуклыми, никакие три вершины одного многоугольника не лежат на одной прямой. Гарантируется, что можно осуществить параллельный перенос многоугольников таким образом, что второй окажется строго внутри первого, и их границы не будут иметь общих точек.

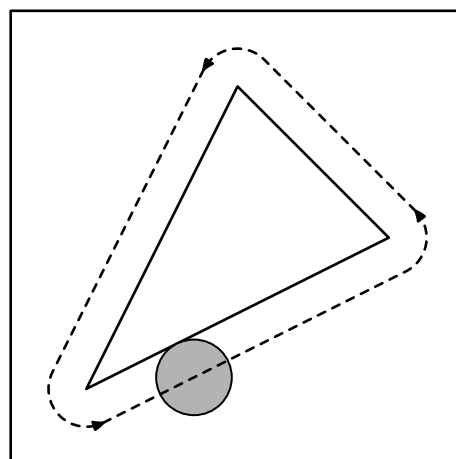
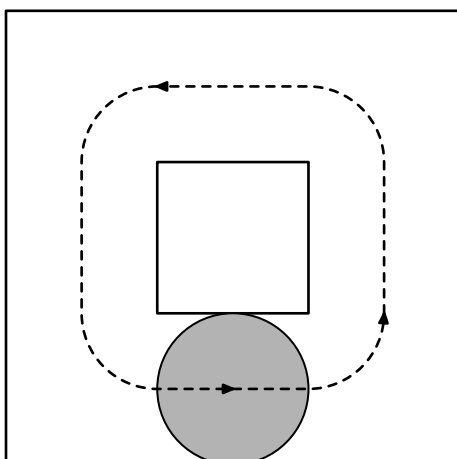
Формат выходных данных

Выведите одно число — максимальный радиус болида. Ответ будет считаться правильным, если его абсолютная или относительная погрешность не превышает 10^{-6} .

Примеры

стандартный ввод	стандартный вывод
4 0 0 3 0 3 3 0 3 4 0 0 1 0 1 1 0 1	0.5
4 0 0 3 0 3 3 0 3 3 0 0 2 1 1 2	0.25

Замечание



Задача L. 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.

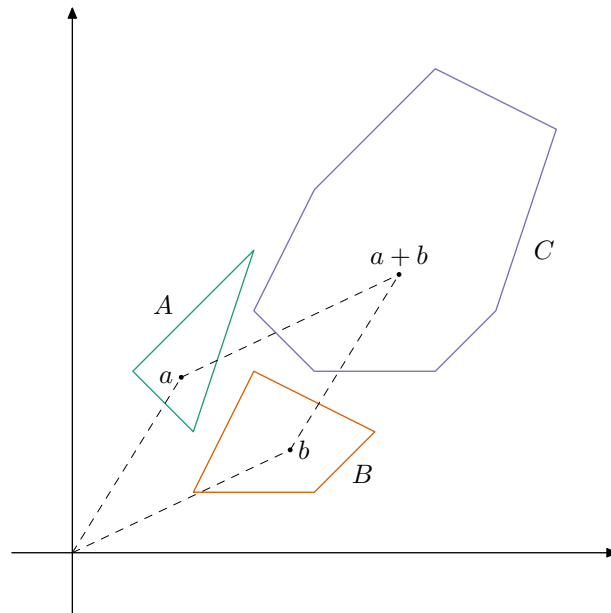


Рис. 1: Сумма двух выпуклых многоугольников

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

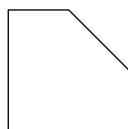


(b) 2-й многоугольник

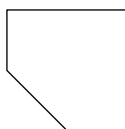


(c) 3-й многоугольник

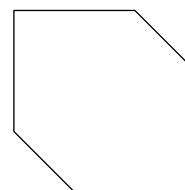
Рис. 2: Многоугольники, которые заготовил Денис



(a) 1 + 2



(b) 2 + 3



(c) 1 + 2 + 3

Рис. 3: Суммы Минковского многоугольников

Задача М. Название

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

Вы знаете разницу между отелем и мотелем? Вы правы, разница в количестве мух, живущих там. Петя владелец одного из самых популярных мотелей в Берляндии, но его мама настаивает на том, чтобы он превратил свой мотель в отель. Именно поэтому она подарила Пете мухобойку в виде многоугольника из k вершин.

Подойдя к окну Петя увидел n мух. Так как Петя и мухи не обидит, он хочет узнать количество способов ударить мухобойкой, не покалечив ни одной мухи.

Окно представляет из себя прямоугольник, нижний левый угол которого находится в центре координатной системы. После удара Пети все вершины мухобойки должны находиться в целых координатах и не должны вылезать за границы окна. Муха покалечена, если она будет находиться под мухобойкой либо на её границах.

Формат входных данных

Первая строка входных данных содержит числа x_p , y_p и n ($1 \leq x_p, y_p \leq 500$, $0 \leq n \leq x_p \cdot y_p$) — координаты правого верхнего угла окна и количество мух на ней, соответственно.

Следующие n строк содержат по два числа x и y ($0 < x < x_p$, $0 < y < y_p$) — координаты мух на окне.

Следующая строка содержит число k ($3 \leq k \leq 10\,000$) — количество вершин у мухобойки. В следующих k строках даются x_i и y_i ($-10^9 \leq x_i, y_i \leq 10^9$) — координаты i -й вершины мухобойки. Вершины заданы в порядке обхода, так что соседние вершины и первая и последняя вершины соединены прямой линией.

Формат выходных данных

Выведите количество способов ударить мухобойкой так, чтобы ни одна муха не пострадала.

Система оценки

Подзадача 1 (30 баллов): $x_p \cdot y_p \cdot n \cdot k \leq 3 \cdot 10^7$ и $x_p, y_p \leq 100$

Подзадача 2 (20 баллов): $x_p, y_p \leq 100$

Подзадача 3 (20 баллов): мухобойка является прямоугольником со сторонами параллельными осям координат

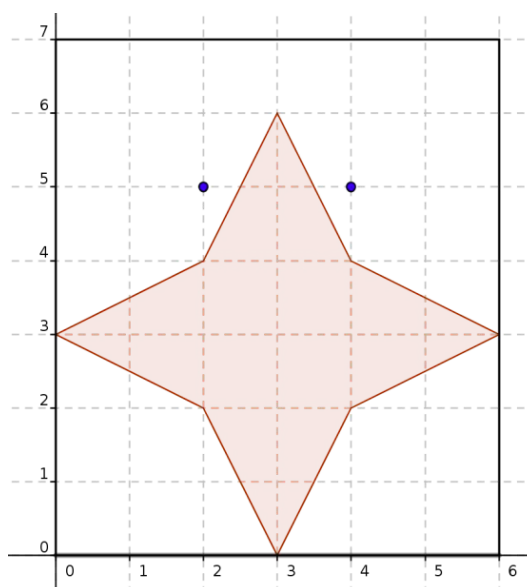
Подзадача 4 (30 баллов): нет дополнительных ограничений

Примеры

стандартный ввод	стандартный вывод
4 5 2 1 3 3 4 4 0 0 2 0 2 2 0 2	4
5 5 3 1 4 1 3 2 2 3 4 7 6 3 7 6	3
6 7 2 2 5 4 5 8 1 4 3 3 4 1 5 3 7 4 5 5 4 7 3 5	1

Замечание

Пояснение к третьему примеру:



Задача N. 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 О. Астрономия

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	5 секунд
Ограничение по памяти:	512 мегабайт

18 год нашей эры. Известный астроном Филон Берляндский публикует трактат «Об устройстве небесного свода», в котором рассказывает о невероятном явлении, замеченном им во время наблюдений за звёздами. Однажды ночью Филон увидел на безоблачном небе $2n$ звёзд и Луну. Удивительно, что звёзды можно было мысленно разбить на пары так, что любая прямая, проходящая через центры двух звёзд из пары проходила также и через центр Луны, при этом все такие прямые были различны. Филон тщательно запечатлел данное явление на карте звёздного неба, в которой была введена система координат, и обнаружил, что центры всех звёзд и центр Луны оказались в точках с целочисленными координатами. Поскольку Филон считал, что Земля и Луна плоские, система координат на карте была двумерная. Система координат была выбрана астрономом так, чтобы координаты всех объектов, включая Луну, по модулю не превосходили 10^6 . Помимо этого, никакие два объекта (две звезды или звезда и Луна) не были в одной точке.

Помимо карты звёздного неба Филон Берляндский написал в своём трактате предсказание, что спустя 2000 лет звёзды на небе вернутся в то же положение, а на месте Луны появится огромная комета, которая уничтожит Землю.

2018 год нашей эры. Вам в руки попадает трактат Филона Берляндского, и вы с ужасом обнаруживаете, что звёзды на небе находятся в том же положении, что и 2000 лет назад! К сожалению, время не пощадило карту астронома, поэтому на ней остались отмеченными только точки, соответствующие центрам звёзд, и не осталось никаких упоминаний, каким образом следовало разбить точки на пары, чтобы все построенные по этим парам прямые прошли через центр Луны. Что ещё хуже, на карте стёрлась точка, соответствующая центру Луны. Чтобы узнать, откуда прилетит комета, и спасти человечество от неминуемой гибели, необходимо срочно восстановить какую-нибудь подходящую позицию центра Луны!

Формат входных данных

В первой строке задано целое число n ($2 \leq n \leq 2600$) — количество пар звёзд, которые астроном увидел на небе.

В следующих $2n$ строках содержатся пары целых чисел x_i, y_i ($-10^6 \leq x_i, y_i \leq 10^6$) — координаты центров звёзд на карте. Обратите внимание, что звёзды указаны в произвольном порядке, никак не связанном с тем, как Филон Берляндский разделял их на пары. Центры никаких двух звёзд не находятся в одной точке.

Формат выходных данных

Если астроном ошибся, и не существует способа разбить все точки на пары таким образом, чтобы все построенные по этим прямые были различны и пересекались в одной точке с целыми координатами, отличной от центров всех звёзд, то выведите «No» (без кавычек) в единственной строке вывода.

В противном случае в первой строке выведите «Yes» (без кавычек). Во второй строке выведите пару целых чисел x, y ($|x|, |y| \leq 10^6$) — координаты точки, в которой находился центр Луны в вашем решении. Если подходящих точек несколько, то выведите любую из них. Обратите внимание, что выведенная точка не может совпадать с центром ни одной из звёзд.

Примеры

стандартный ввод	стандартный вывод	Пояснение
2 1 1 1 3 3 1 3 3	Yes 2 2	
3 4 2 6 2 2 1 2 6 4 4 6 6	Yes 2 2	
2 1 1 2 2 4 4 5 5	No	
2 1 1 2 1 1 2 2 2	No	
2 0 0 1 1 -1 -1 -1 1	No	

Замечание

В четвёртом тесте из условия центр Луны мог располагаться только в точке $(1.5, 1.5)$, но эта точка обладает нецелыми координатами, поэтому ответа нет.

В пятом тесте из условия невозможно найти подходящую точку, которая бы не являлась при этом центром звезды.

Задача Р. 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