

## Задача А. Платные дороги

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

Мэр одного большого города решил ввести плату за проезд по шоссе, проходящим в районе города, чтобы снизить объем транзитного транспорта. В районе города проходит  $n$  шоссе.

Но руководство области, в которой расположен город, воспротивилось планам мэра. Действительно — дальнбойщики представляют собой неплохой источник доходов для большого количества кафе и гостиниц в небольших городках.

В результате решили, что плата будет введена только на шоссе, которые проходят через город.

В городе используется развитая система метрополитена, всего в городе есть  $m$  станций метро. Решено было, что шоссе проходит через город, если либо одна из станций метро расположена непосредственно на шоссе, либо есть хотя бы одна станция с каждой стороны от шоссе.

Помогите теперь мэру определить, какие шоссе проходят через город.

### Формат входных данных

Первая строка входного файла содержит два целых числа:  $n$  и  $m$  — количество шоссе и количество станций метро, соответственно ( $1 \leq n, m \leq 100\,000$ ).

Следующие  $n$  строк описывают шоссе. Каждое шоссе описывается тремя целыми числами  $a$ ,  $b$  и  $c$  и представляет собой прямую на плоскости, задаваемую уравнением  $ax + by + c = 0$  ( $|a|, |b|, |c| \leq 10^9$ ).

Следующие  $m$  строк входного файла описывают станции метро. Каждая станция описывается двумя целыми числами  $x$  и  $y$  и представляет собой точку на плоскости с координатами  $(x, y)$  ( $|x|, |y| \leq 10^9$ ).

### Формат выходных данных

Первая строка выходного файла должна содержать одно целое число — количество шоссе, которые проходят через город. Вторая строка должна содержать номера этих шоссе в возрастающем порядке. Шоссе нумеруются от 1 до  $n$  в порядке, в котором они описаны во входном файле.

### Примеры

стандартный ввод	стандартный вывод
4 2 0 1 0 1 0 1 1 1 0 1 1 -1 0 0 2 0	3 1 3 4
12 1 0 1 0 0 1 1 0 1 -1 1 0 1 1 0 -1 1 0 0 1 1 0 1 1 1 1 1 -1 1 -1 0 1 -1 1 1 -1 -1 0 0	4 1 6 7 10

## Задача В. 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

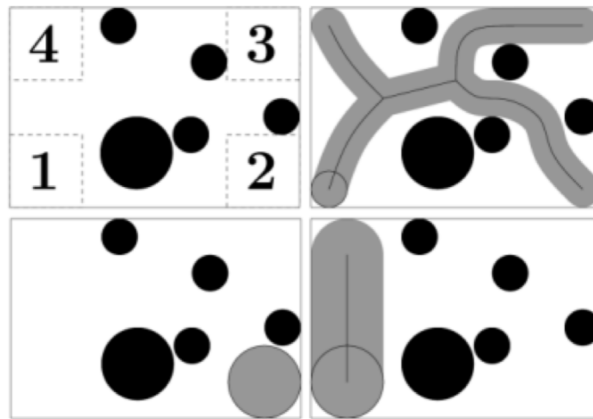
Вы должны вывести для каждого посетителя одну строку, содержащую входы, через которые они могут выйти из парка, в отсортированном порядке без пробелов между ними.

### 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 С. 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	1.00000000000000000000 1.00000000000000000000
0 2	1.41421356237309510000
0 0	
2 0	

## Задача D. 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 Е. Силовое поле

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	3 секунды
Ограничение по памяти:	256 мегабайт

Геометрия? Интерактивная?

Империя обнаружила мятежников на ледяной планете Хот! По сведениям разведки все командование Альянса Повстанцев сейчас скрывается на базе «Эхо», спрятанной в горах на севере этой суровой планеты.

Для того, чтобы окончательно подавить силы восстания, необходимо в ходе стремительной атаки уничтожить эту базу и скрывающихся на ней мятежников. К сожалению, укрытие хорошо укреплено: в частности, его защищает мощное силовое поле, препятствующее бомбардировкам с орбиты. Силовое поле имеет форму выпуклого многоугольника с вершинами в  $N$  специальных станциях-ретрансляторах. Никакие три станции не располагаются на одной прямой.

Перед тем как начинать операцию по уничтожению повстанцев, требуется лишить их базу силового поля, уничтожив эти  $N$  станций точечным бомбометанием. Однако точные координаты этих станций нам неизвестны. Ваша цель — узнать расположение станций- ретрансляторов, чтобы наши войска смогли начать наступление.

На планете введена система координат, устроенная так, что все станции-ретрансляторы находятся в точках с целыми координатами, не превосходящими  $C$  по модулю.

В вашем распоряжении есть зонд-разведчик, оснащенный специальным оборудованием, позволяющим регистрировать станции-ретрансляторы. Если запустить его по прямой над базой повстанцев, по его информации можно будет узнать, сколько станций-ретрансляторов располагаются слева, и сколько — справа от прямой его движения. Станции, находящиеся на его пути, зонд не регистрирует.

С повстанцами надо расправиться как можно скорее: у вас есть время не более чем на  $10^5$  запусков этого зонда. Восстановите по полученной от него информации точные координаты станций-ретрансляторов, чтобы мы могли начать наступление, и Империя вас не забудет!

### Протокол взаимодействия

При запуске решения на вход подаются два целых числа  $N$  и  $C$  — количество станций ( $3 \leq N \leq 1\,000$ ) и ограничение на абсолютную величину их координат ( $5 \leq C \leq 1\,000\,000$ ).

Для запуска зонда выведите строку «?  $x_1\ y_1\ x_2\ y_2$ », где  $(x_1, y_1)$  и  $(x_2, y_2)$  — две точки с целочисленными координатами, лежащие на прямой, по которой должен лететь зонд.

Зонд будет лететь в направлении от первой точки ко второй. Точки не должны совпадать. Координаты точек не должны превосходить  $5C$  по модулю.

На каждый запуск зонда-разведчика вводится полученная им информация — два целых числа  $l$  и  $r$ , разделенных пробелом, — количество станций-ретрансляторов слева и справа от прямой его движения соответственно.

Как только вы найдете ответ, выведите строку «Ready!» и в следующих  $N$  строках выведите координаты станций в любом порядке. После этого ваша программа должна завершиться.

## Пример

стандартный ввод	стандартный вывод
4 5	? -1 3 1 3
0 4	? -1 2 1 2
0 3	? -1 1 0 2
0 3	? -1 0 0 2
0 2	? 0 0 0 2
1 1	? 1 0 1 2
3 1	? 2 0 2 2
3 0	? 3 0 1 2
3 0	Ready!
	0 -1
	2 1
	0 2
	-1 0

## Замечание

В точности соблюдайте формат выходных данных. После вывода каждой строки сбрасывайте буфер вывода.

Программа не должна делать более  $10^5$  запросов запуска зонда. При превышении этого количества, тест будет не пройден с вердиктом «Wrong Answer».

## Задача F. Тигры Easy

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  $n \leq 25000$   $k = 200$ .

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.

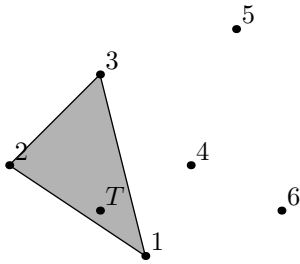
Ответное сообщение от программы жюри отсутствует, и программа-решение должна немедленно приступить к поиску следующего тигра. Локализовав положение тигра с номером  $q$ , программа-решение должна завершить работу.

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

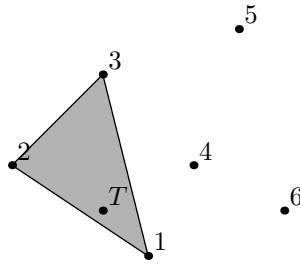
Если существует несколько правильных многоугольников, локализующих положение тигра, можно вывести любой из них.

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

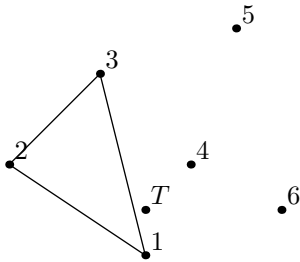
Тигр 1, запрос 1:



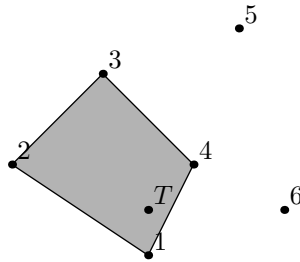
Тигр 1, ответ:



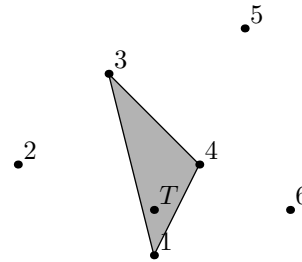
Тигр 2, запрос 1:



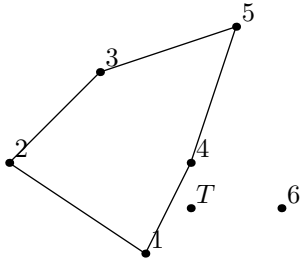
Тигр 2, запрос 2:



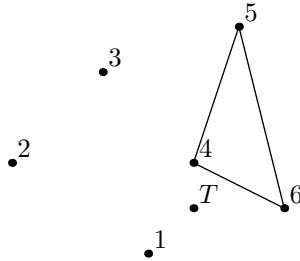
Тигр 2, ответ:



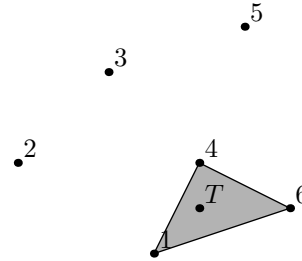
Тигр 3, запрос 1:



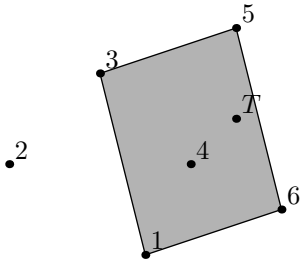
Тигр 3, запрос 2:



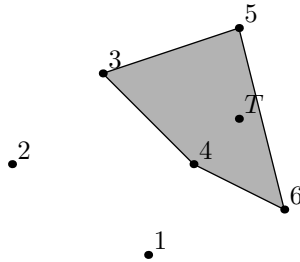
Тигр 3, ответ:



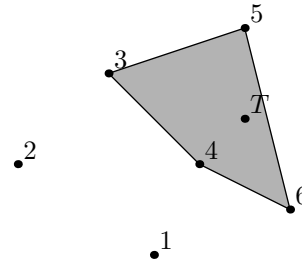
Тигр 4, запрос 1:



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 G. Триангуляция многоугольника

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

Дан простой невырожденный, возможно невыпуклый, возможно с развернутыми углами,  $N$ -угольник. Вам нужно найти его триангуляцию.

### Формат входных данных

В первой строке число  $N$  ( $3 \leq N \leq 4000$ ) — количество вершин. Далее  $N$  строк, содержащие пары целых чисел, — координаты вершин многоугольника. Все координаты целые, по модулю не превосходят  $10^4$ .

### Формат выходных данных

Выведите  $N - 3$  диагонали. Каждая задается парой чисел от 0 до  $N - 1$  — номера вершин.

Отрезок  $(i, j)$  считается диагональю, если вся его внутренность лежит строго внутри многоугольника.

### Примеры

стандартный ввод	стандартный вывод
3 0 0 1 0 1 1	
4 0 0 1 0 1 1 0 1	3 1

## Задача Н. Диаметр точек

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 64 мегабайта

На плоскости даны  $N$  точек. Вам требуется найти расстояние между двумя самыми удаленными точками.

### Формат входных данных

Первая строка содержит количество точек  $N$ , ( $1 \leq N \leq 10^5$ ). Каждая из последующих  $N$  строк содержит два целых числа — координаты  $x_i$  и  $y_i$ . Координаты по модулю не превосходят  $10^9$ .

### Формат выходных данных

Выведите в выходной файл расстояние между двумя наиболее удалёнными точками с максимально возможной точностью.

### Примеры

стандартный ввод	стандартный вывод
5 0 0 2 2 1 1 0 2 2 0	2.828427124746
7 0 0 1 1 2 2 0 2 1 3 0 1 2 0	3.162277660168

## Задача I. 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