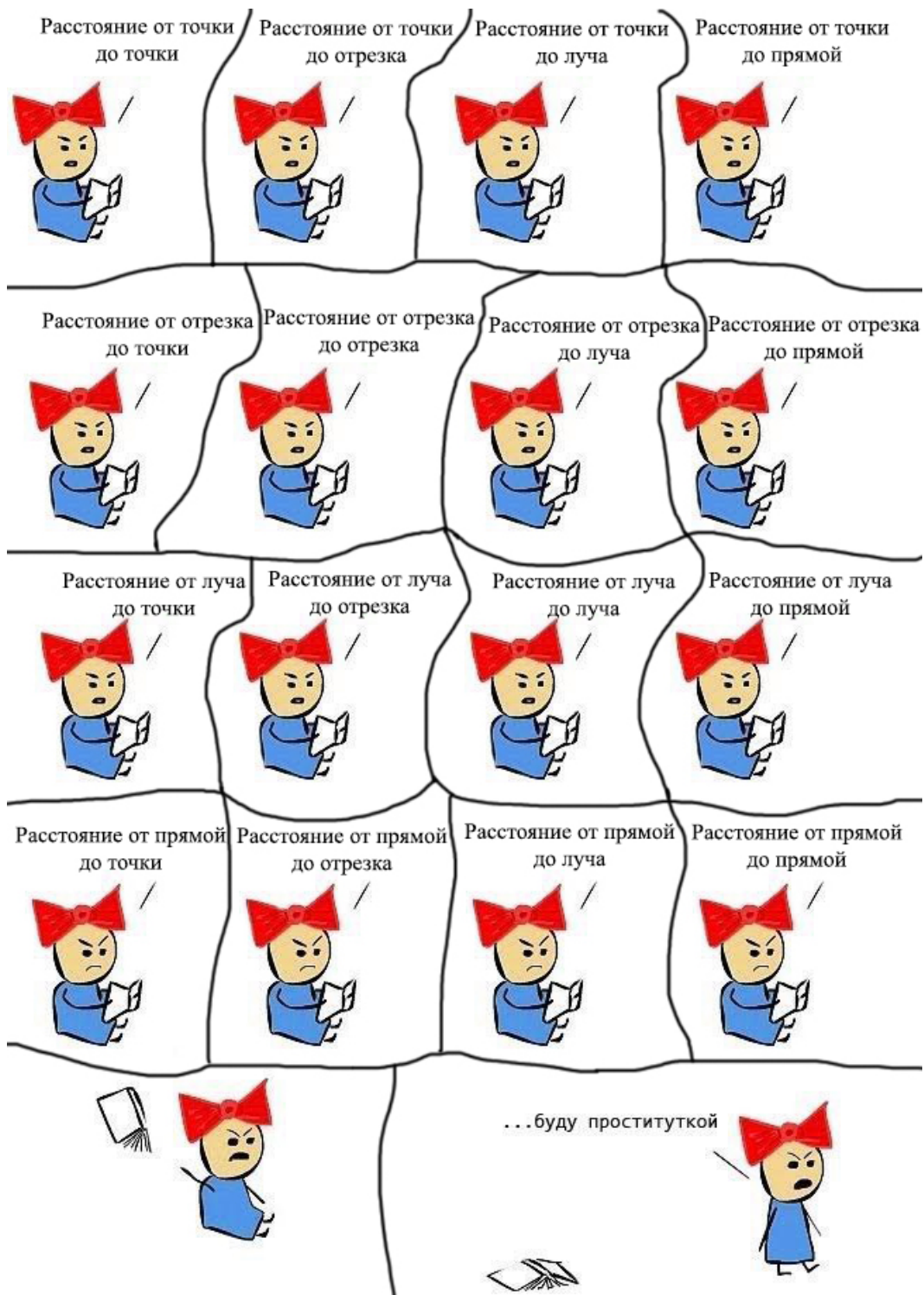


## Задача А. 16

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. Периметр выпуклой оболочки

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

Дано  $N$  точек на плоскости.

Нужно вычислить периметр выпуклой оболочки данных точек.

Гарантируется, что выпуклая оболочка не вырождена.

### Формат входных данных

В первой строке число  $N$  ( $3 \leq N \leq 10^5$ ). Следующие  $N$  строк содержат пары целых чисел  $x$  и  $y$  ( $-10^9 \leq x, y \leq 10^9$ ) — точки.

Будьте аккуратны! Точки произвольны. Бывают совпадающие, бывают лежащие на одной прямой в большом количестве.

### Формат выходных данных

Одно вещественное число — периметр выпуклой оболочки. Выводите число с максимально возможной точностью.

### Пример

стандартный ввод	стандартный вывод
5 0 0 2 0 0 2 1 1 2 2	8

## Задача С. Замок

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

Есть замок — точка  $(0, 0)$ . Замок окружен несколькими непересекающимися заборами, каждый представляет из себя выпуклый многоугольник.

Есть также  $m$  захватчиков, известны их координаты. Захватчики не умеют перелезть через заборы. Захватчика будем считать опасным, если он находится внутри внешнего забора. Требуется вычислить суммарную площадь области, куда опасные захватчики могут добраться без пересечения заборов.

### Формат входных данных

В первой строке задано число  $n$  — количество заборов ( $1 \leq n \leq 100\,000$ ). Далее следуют описания  $n$  заборов. Каждое описание начинается с числа  $k$ , далее следуют  $k$  строк, содержащих по два числа  $x$  и  $y$  — координаты вершин ( $|x|, |y| \leq 2 \cdot 10^6$ ). Вершины каждого многоугольника перечисляются в порядке обхода против часовой стрелки. Гарантируется, что точка  $(0, 0)$  лежит внутри каждого забора.

Далее следует число  $m$  ( $0 \leq m \leq 100\,000$ ) — количество захватчиков. В следующих  $m$  строках заданы координаты захватчиков.

Суммарное число вершин во всех многоугольниках не превосходит  $100\,000$ .

### Формат выходных данных

Выведите единственное число — общую захваченную площадь с шестью знаками после десятичной точки.

### Пример

стандартный ввод	стандартный вывод
3	2400.000000
4	
-10 -10	
10 -10	
10 10	
-10 10	
4	
20 20	
-20 20	
-20 -20	
20 -20	
4	
30 -30	
30 30	
-30 30	
-30 -30	
3	
1 1	
22 23	
111 123	

## Задача D. 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 Е. Извилистая ломаная

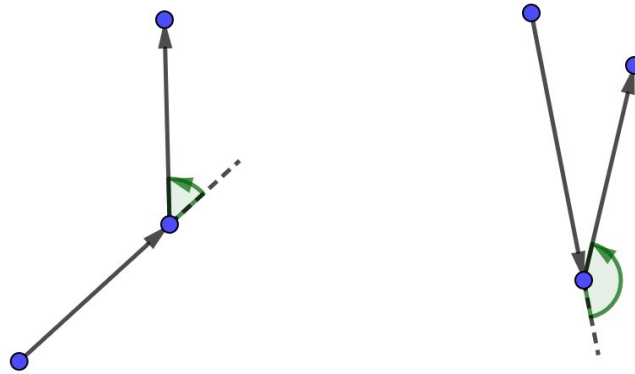
Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

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  $n$ .

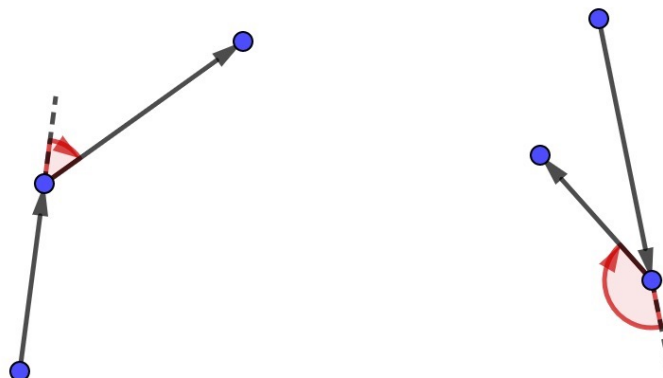
Сделав так, он нарисует ориентированную ломаную на этих вершинах, проведя направленные отрезки из каждой точки в следующую в выбранном порядке точку. То есть для всех  $1 \leq i \leq n-1$  он проведет направленный отрезок из точки  $A_{p_i}$  в точку  $A_{p_{i+1}}$ . Он хочет, чтобы получившаяся ломаная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

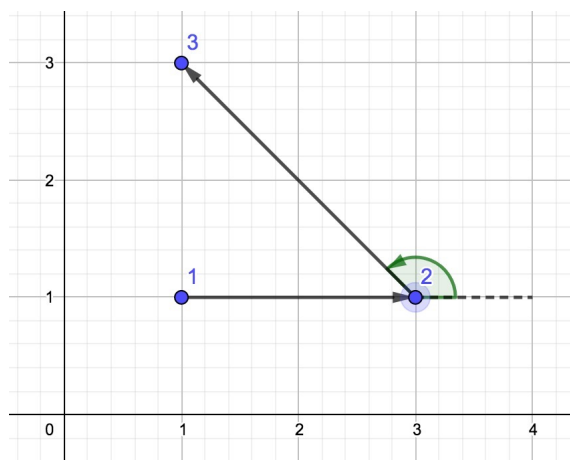
Если подходящей перестановки не существует выведите  $-1$ . Иначе выведите  $n$  чисел  $p_1, p_2, \dots, p_n$  — найденную перестановку ( $1 \leq p_i \leq n$  и все  $p_1, p_2, \dots, p_n$  различны). Если подходящих перестановок несколько, выведите любую.

### 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

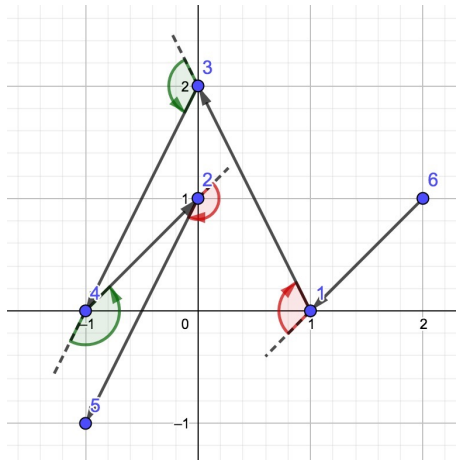
### Замечание

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 F. 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

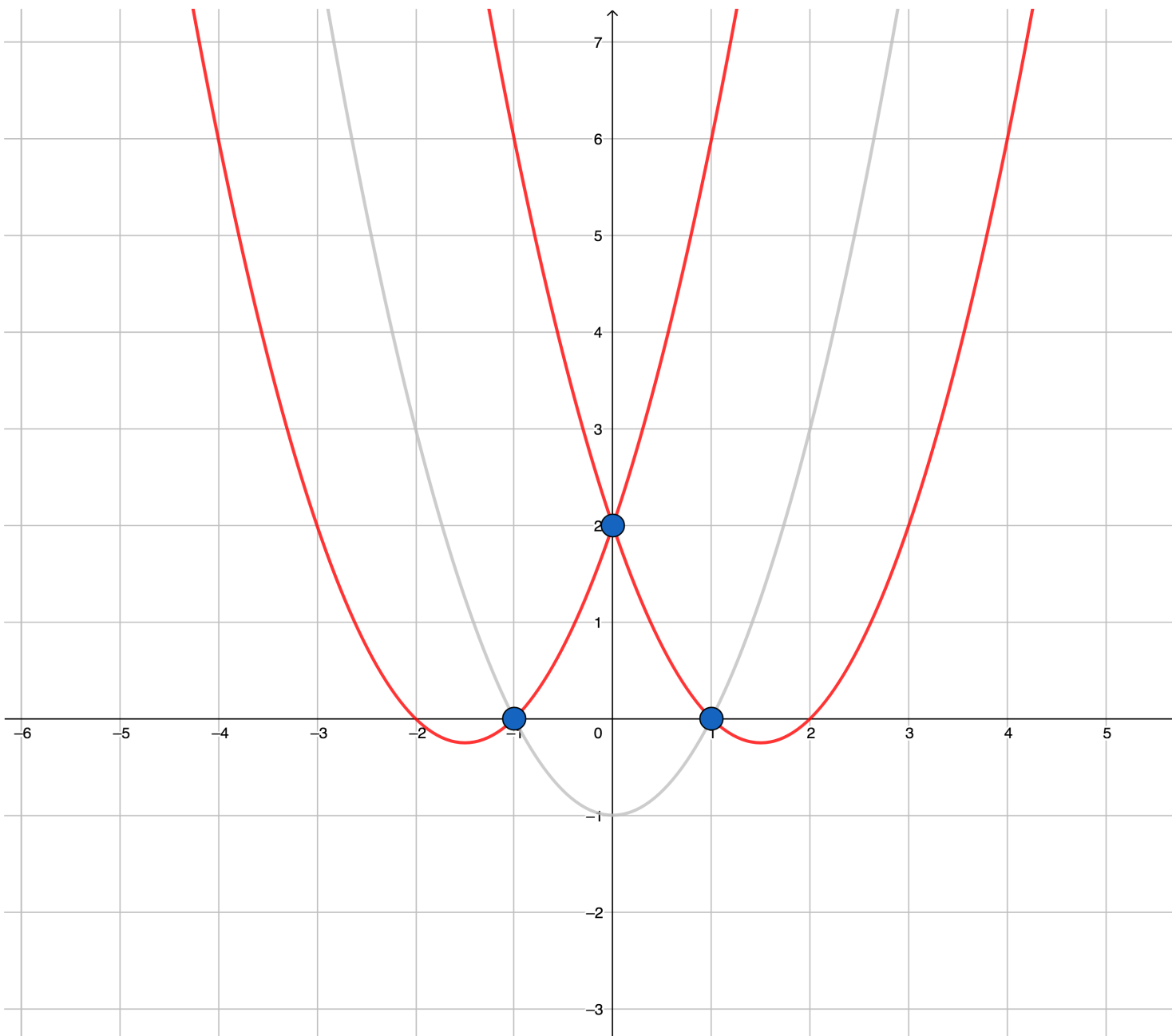
В единственной строке выведите единственное число — количество  $U$ -образных парабол, проходящих хотя бы через 2 точки и не содержащих никаких других точек в своей внутренней области (не считая границы).

### Примеры

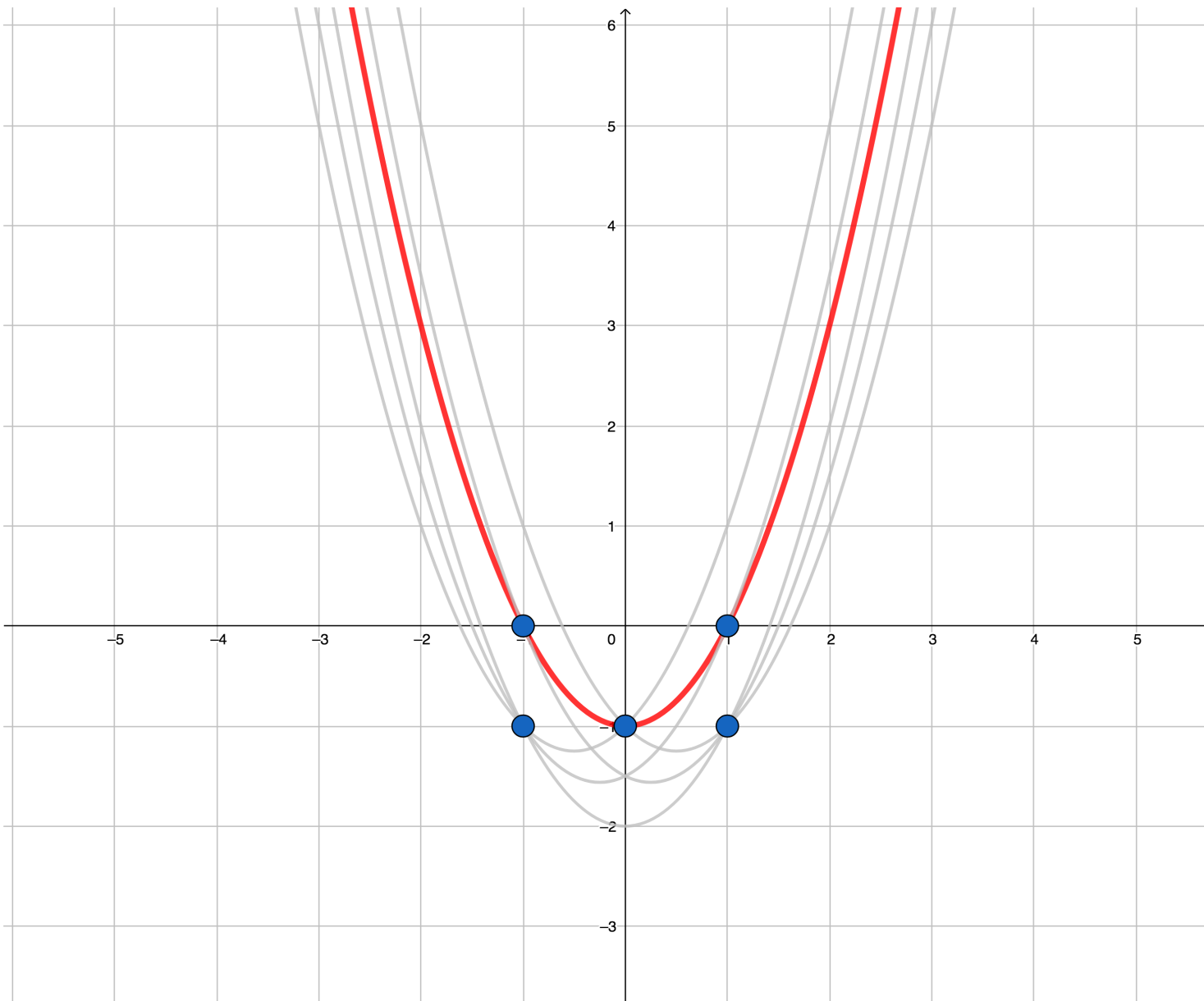
стандартный ввод	стандартный вывод
3 -1 0 0 2 1 0	2
5 1 0 1 -1 0 -1 -1 0 -1 -1	1

### 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 G. Место встречи изменить нельзя

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 0.3 секунд  
Ограничение по памяти: 256 мегабайт

Даны  $N$  точек. Найдите такие две из них, что расстояние между ними минимально.

### Формат входных данных

Первая строка входного файла содержит целое число  $N$  ( $2 \leq N \leq 100\,000$ ) — количество точек. Каждая из следующих  $N$  строк содержит пару целых чисел  $X$  и  $Y$ , разделённых пробелом, — координаты ( $-1\,000\,000\,000 \leq X, Y \leq 1\,000\,000\,000$ ). Все точки различны.

### Формат выходных данных

Единственная строка выходного файла должна содержать координаты двух выбранных точек.

### Пример

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

## Задача Н. 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

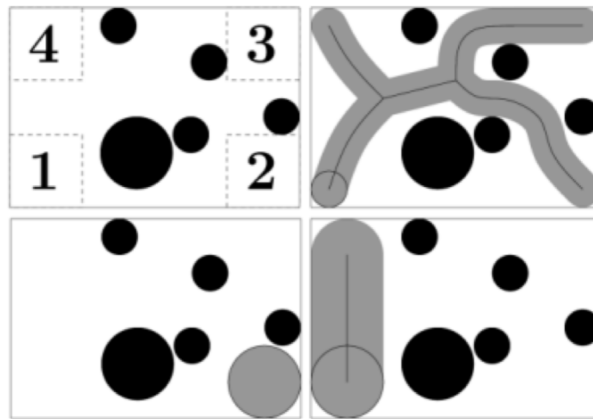
Вы должны вывести для каждого посетителя одну строку, содержащую входы, через которые они могут выйти из парка, в отсортированном порядке без пробелов между ними.

### 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 I. 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 J. 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 К. 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  $\alpha_j$  раз по  $x$ -координате.

Более формально, у Игоря есть  $q$  запросов, в  $j$ -м из которых для числа  $\alpha_j$  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  $\alpha_j$  ( $1 \leq \alpha_j \leq 10^9$ ) — 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

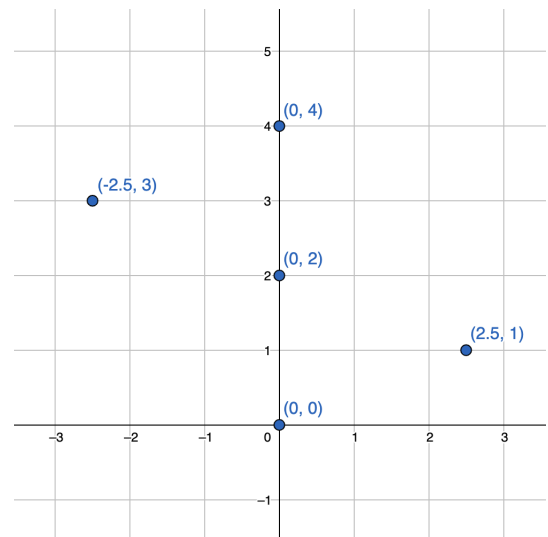
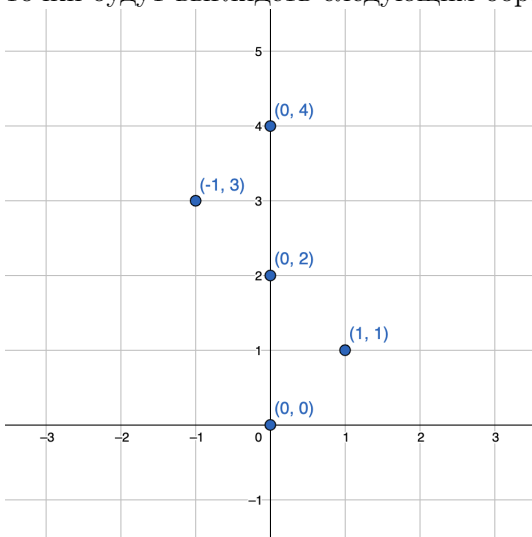
Для каждого набора входных данных выведите  $q$  строк, в  $i$ -й строке должно содержаться единственное вещественное число — ответ на  $i$ -й запрос. Ответ будет считаться правильным, если его абсолютная или относительная погрешность не превышает  $10^{-6}$ . Более формально, если  $a$  — ваш ответ, 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 L. Прибытие

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

Радиотелескоп РАТАН-600 засек НЛО на подлете к Земле. НЛО представляет собой две касающиеся окружности неизвестного радиуса  $r$ . По предварительным подсчетам он совершит посадку на участке сибирской тайги. Участок тайги представляет собой прямоугольник размером  $x$  на  $y$ , на котором растут  $n$  ёлок,  $i$ -я ёлка расположена в точке с координатами  $x_i, y_i$ .

НЛО будет посажен так, что все точки на его границе окажутся строго внутри ограничивающего прямоугольника. Ученых интересует максимальный радиус НЛО  $r$  такой, что можно посадить корабль, не повредив ни одной ёлки. Ёлка считается поврежденной, если она оказалась строго внутри одной из окружностей, составляющих корабль.

### Формат входных данных

В первой строке вводится  $t$  ( $1 \leq t \leq 1000$ ) — количество наборов входных данных. В следующих строках содержатся описания каждого тестового набора.

В первой строке каждого описания вводятся два целых числа  $x, y$  ( $2 \leq x, y \leq 10\,000$ ) — размеры ограничивающего прямоугольника.

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

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 ёлок.

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

### Формат выходных данных

Для каждого тестового набора выведите вещественное число  $r$  — максимальный радиус НЛО 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 окружностей, соответствующих ответам на первый тест из условия.

