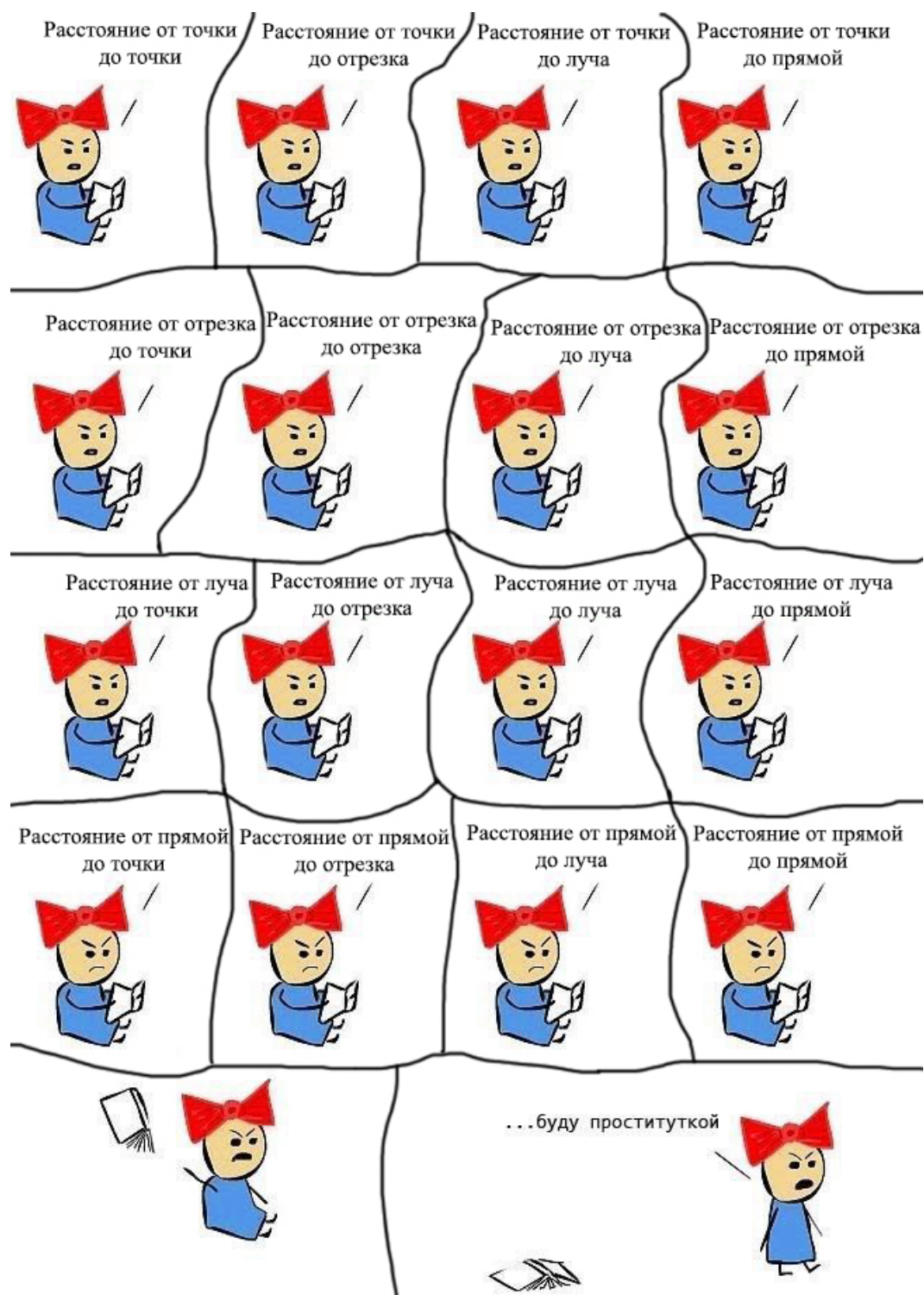


Задача А. 16

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

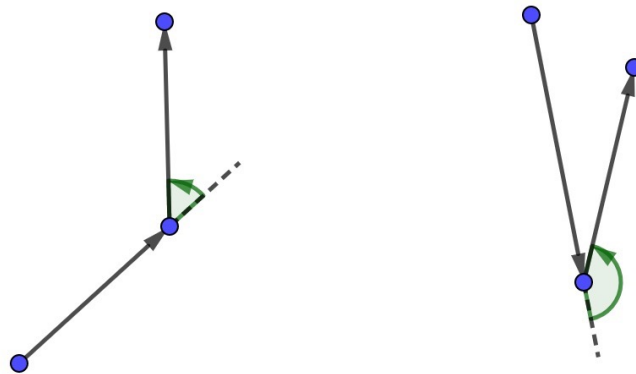
Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

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 .

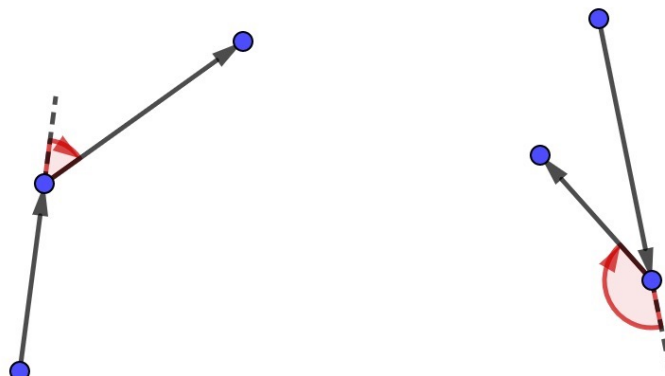
Сделав так, он нарисует ориентированную ломаную на этих вершинах, проведя направленные отрезки из каждой точки в следующую в выбранном порядке точку. То есть для всех $1 \leq i \leq n-1$ он проведет направленный отрезок из точки A_{p_i} в точку $A_{p_{i+1}}$. Он хочет, чтобы получившаяся ломаная удовлетворяла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

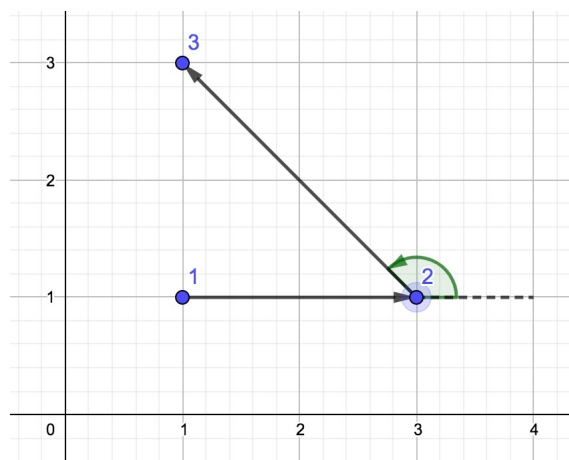
Если подходящей перестановки не существует выведите -1 . Иначе выведите n чисел $p_1, p_2, \dots, p_n$ — найденную перестановку ($1 \leq p_i \leq n$ и все $p_1, p_2, \dots, p_n$ различны). Если подходящих перестановок несколько, выведите любую.

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

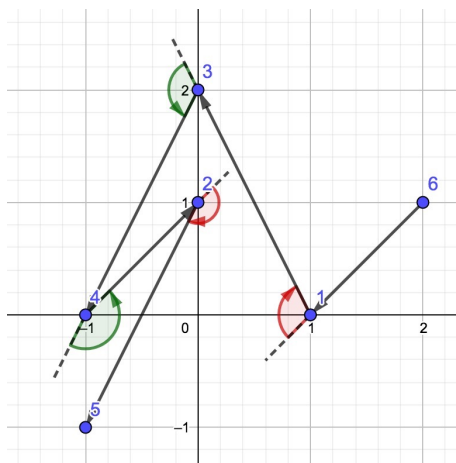
Замечание

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

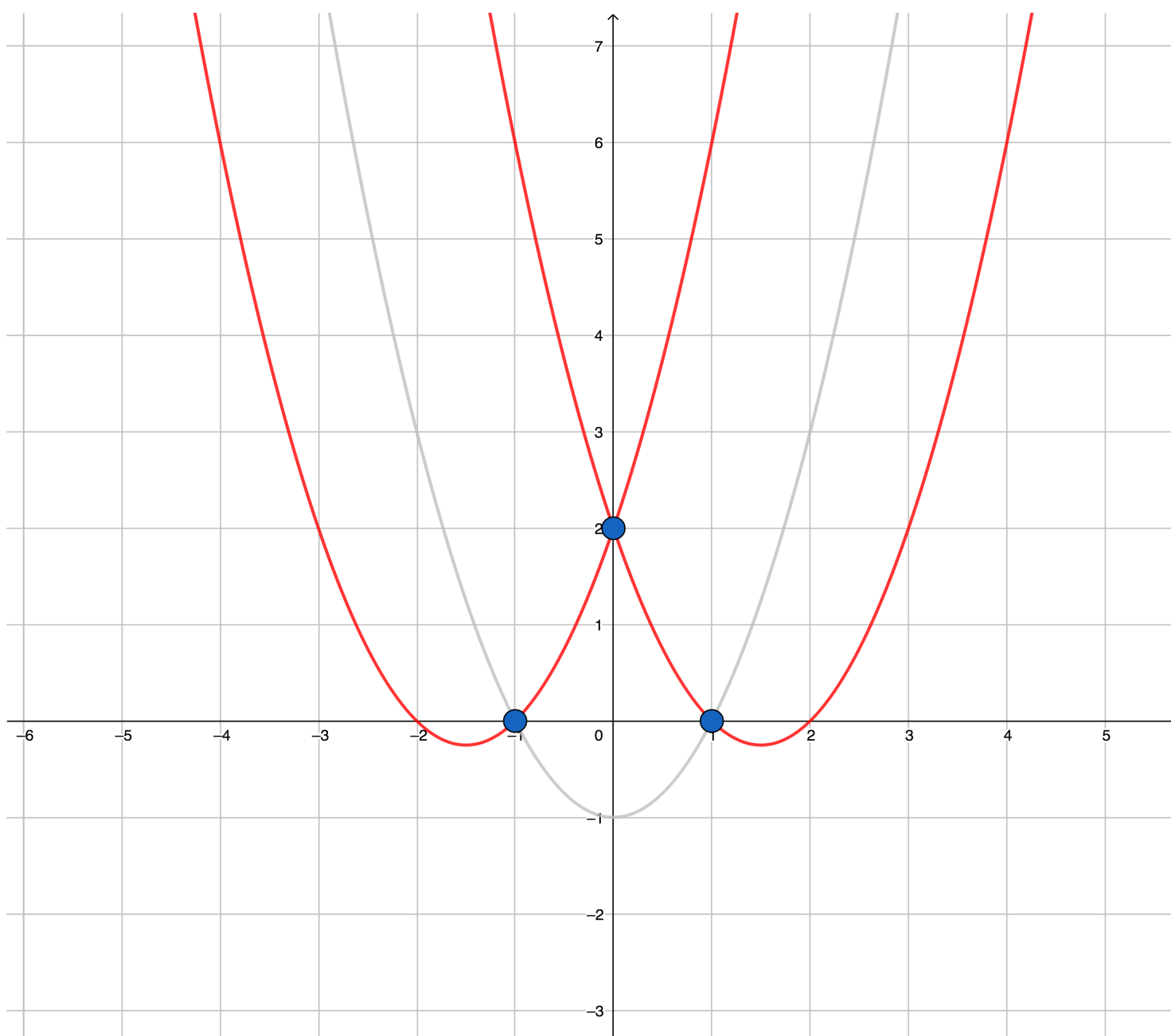
В единственной строке выведите единственное число — количество U -образных парабол, проходящих хотя бы через 2 точки и не содержащих никаких других точек в своей внутренней области (не считая границы).

Примеры

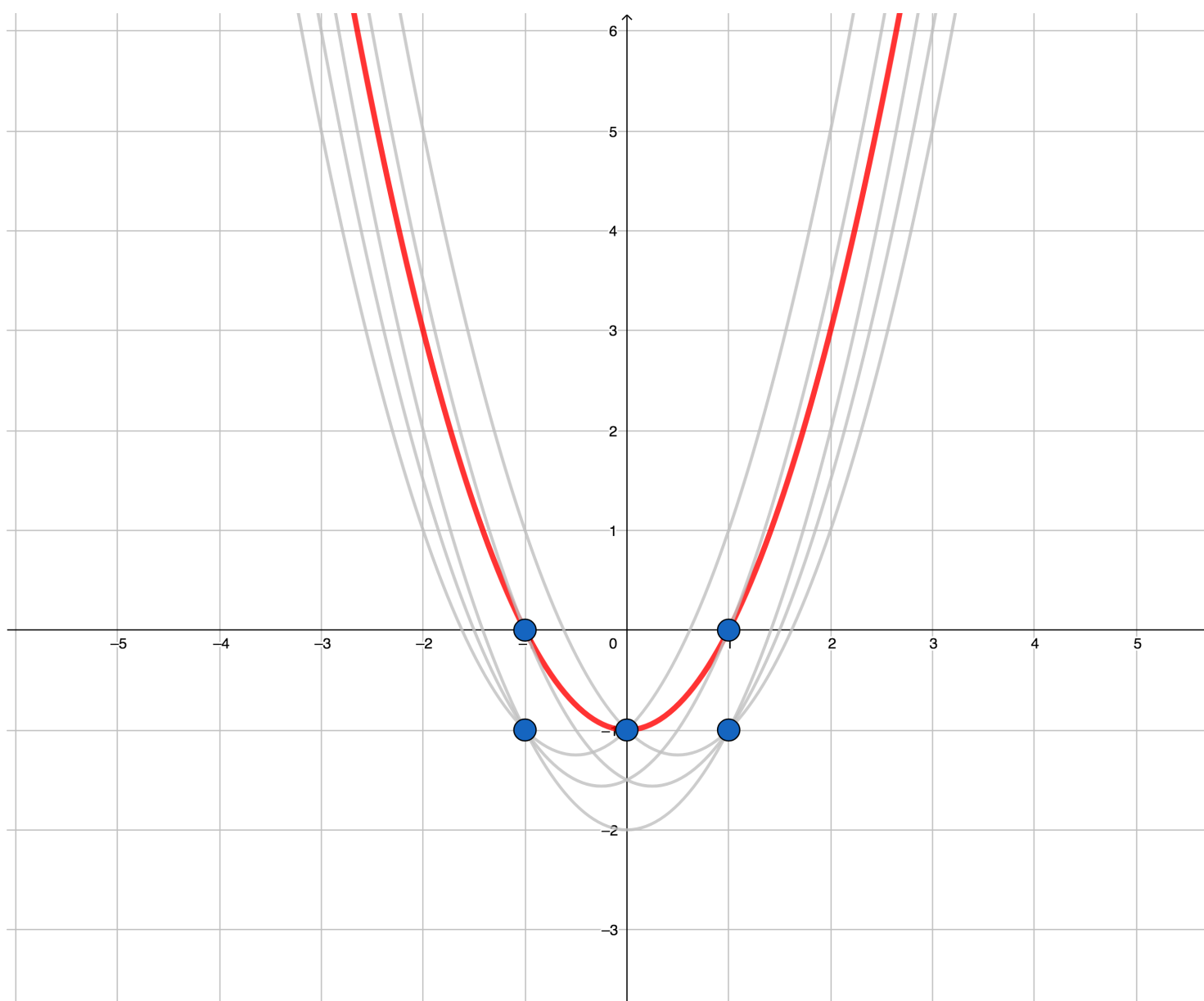
стандартный ввод	стандартный вывод
3 -1 0 0 2 1 0	2
5 1 0 1 -1 0 -1 -1 0 -1 -1	1

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. Женья — очень экономный студент, поэтому он решил купить самую дешевую мухобойку — с минимально возможным радиусом, но Женья 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 G. 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 съемочной площадке не подозревал о том, что Вася в самом деле хочет съесть Петю!

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

Съемочная 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 съемочной 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.

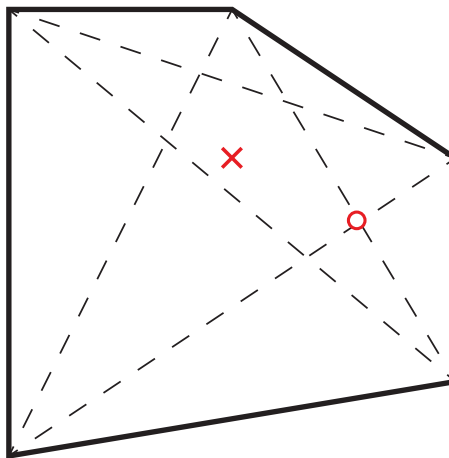
В i -й строке выходного файла выведите одно число — максимальное расстояние, на котором Вася может спрятаться от Пети, когда Петя будет сажать дерево в i -й день. Ваш ответ будет считаться правильным, если его абсолютная или относительная погрешность относительно правильного ответа составит не более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

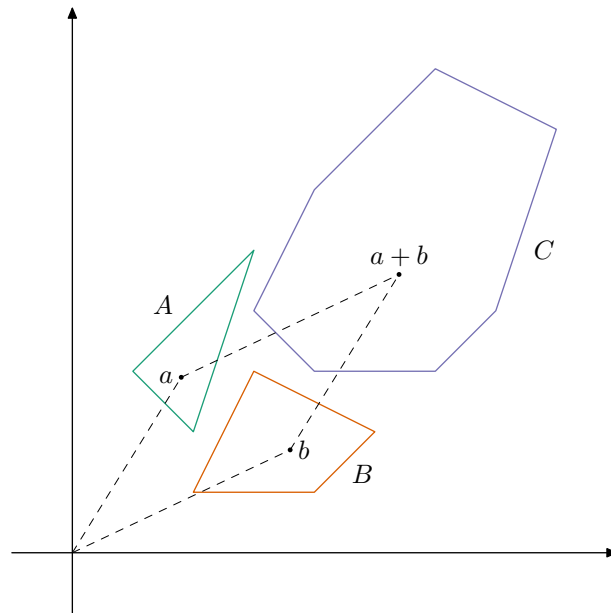


Рис. 1: Сумма двух выпуклых многоугольников

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

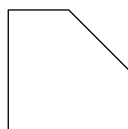


(b) 2-й многоугольник

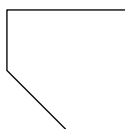


(c) 3-й многоугольник

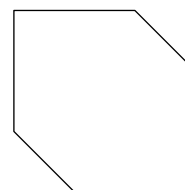
Рис. 2: Многоугольники, которые заготовил Денис



(a) 1 + 2



(b) 2 + 3



(c) 1 + 2 + 3

Рис. 3: Суммы Минковского многоугольников

Задача I. 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 J. Точки на плоскости

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

Даны n точек на плоскости и q запросов следующего вида:

- 1 $x_1 y_1 x_2 y_2$ ($x_1 \neq x_2$ или $y_1 \neq y_2$). Рассмотрим прямую, проходящую через точки (x_1, y_1) и (x_2, y_2) . Существуют две полуплоскости, для которых данная прямая является границей. Рассмотрим ту из них, в которую попадает точка $(x_1 + y_2 - y_1, y_1 + x_1 - x_2)$. Другими словами, если рассмотреть направленный вектор из (x_1, y_1) в (x_2, y_2) , то требуемая полуплоскость будет лежать справа. Вам необходимо проверить, принадлежит ли хотя бы одна из данных точек заданной полуплоскости. Обратите внимание, что прямая принадлежит полуплоскости, поэтому точки, попадающие на прямую, учитываются.
- 2 $x_1 y_1 x_2 y_2$ ($x_1 \neq x_2$ или $y_1 \neq y_2$). Рассмотрим квадрат, противоположными вершинами которого являются точки (x_1, y_1) и (x_2, y_2) . Обратите внимание, что стороны этого квадрата не обязательно параллельны осям координат. Необходимо проверить, есть ли хотя бы одна заданная точка, которая лежит внутри или на границе квадрата.

Формат входных данных

В первой строке заданы два целых числа n и q ($1 \leq n, q \leq 100\,000$) — количество точек и количество запросов.

В следующих n строках содержатся пары целых чисел x_i, y_i ($1 \leq x_i, y_i \leq 10^8$), описывающие координаты точек на плоскости.

В следующих q строках заданы запросы в формате, описанном выше. ($1 \leq x_{i,1}, y_{i,1}, x_{i,2}, y_{i,2} \leq 10^8$)

Формат выходных данных

Для каждого запроса выведите «Yes», если в заданном объекте лежит хотя бы одна точка и «No» — иначе.

Система оценки

Тесты к этой задаче состоят из восьми групп. Баллы за каждую группу ставятся только при прохождении всех тестов группы и всех тестов **необходимых** групп.

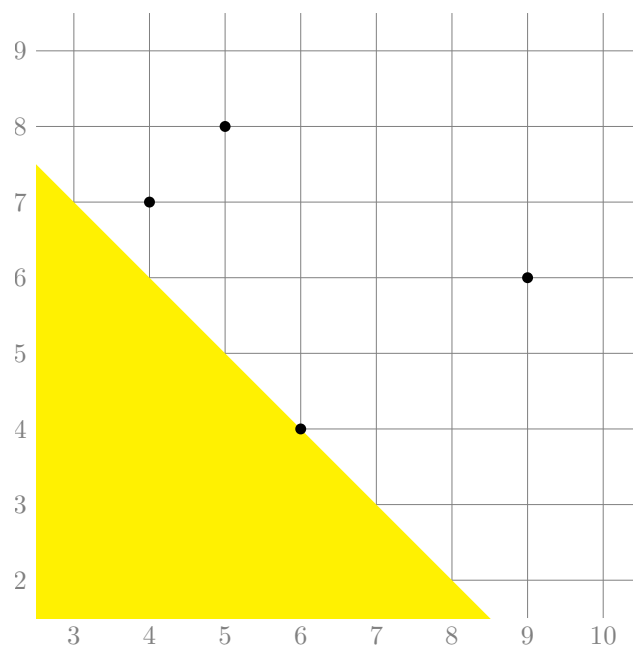
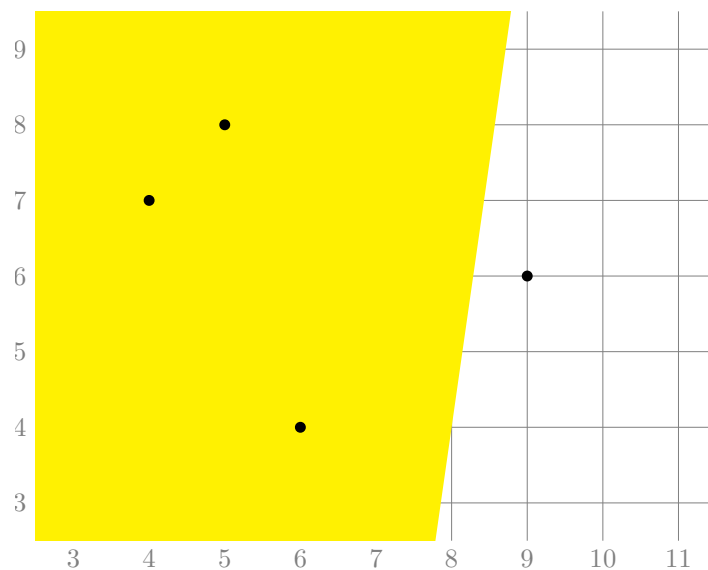
Группа	Баллы	Дополнительные ограничения		Необх. группы	Комментарий
		n	q		
0	0	—	—	—	Тесты из условия.
1	11	$n \leq 1000$	$q \leq 1000$	—	Для всех i верно, что $t_i = 1$.
2	12	—	—	1	Для всех i верно, что $t_i = 1$.
3	5	$n \leq 1000$	$q \leq 1000$	—	Для всех i верно, что $t_i = 2$ и $x_{i,1} - x_{i,2} = y_{i,1} - y_{i,2}$.
4	16	—	—	3	Для всех i верно, что $t_i = 2$ и $x_{i,1} - x_{i,2} = y_{i,1} - y_{i,2}$.
5	12	$n \leq 1000$	$q \leq 1000$	0, 1, 3	
6	17	$n \leq 30\,000$	$q \leq 30\,000$	0, 1, 3, 5	
7	9	$n \leq 60\,000$	$q \leq 60\,000$	0, 1, 3, 5, 6	
8	18	—	—	0 – 7	

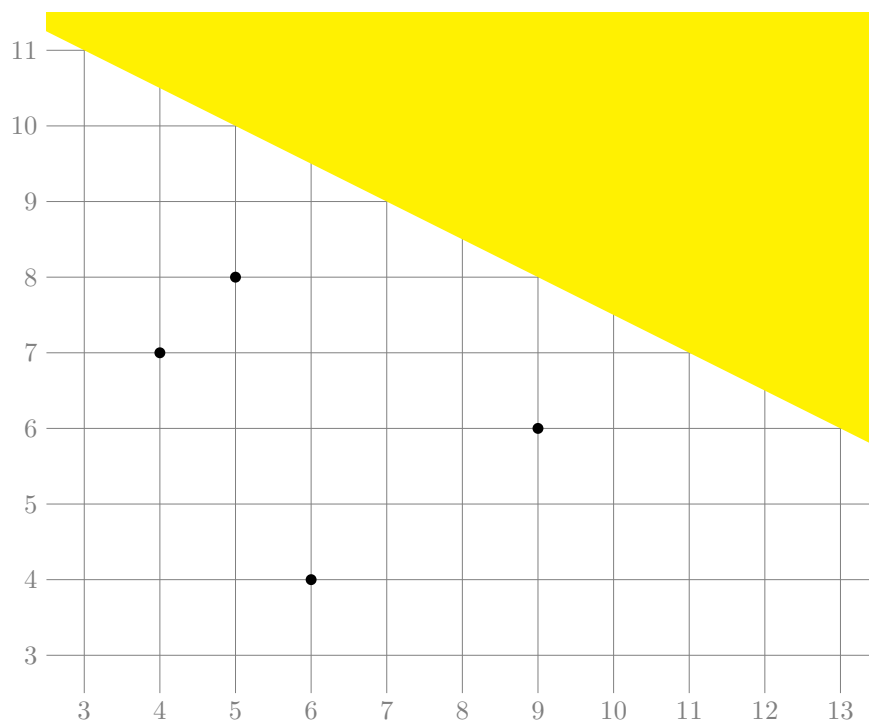
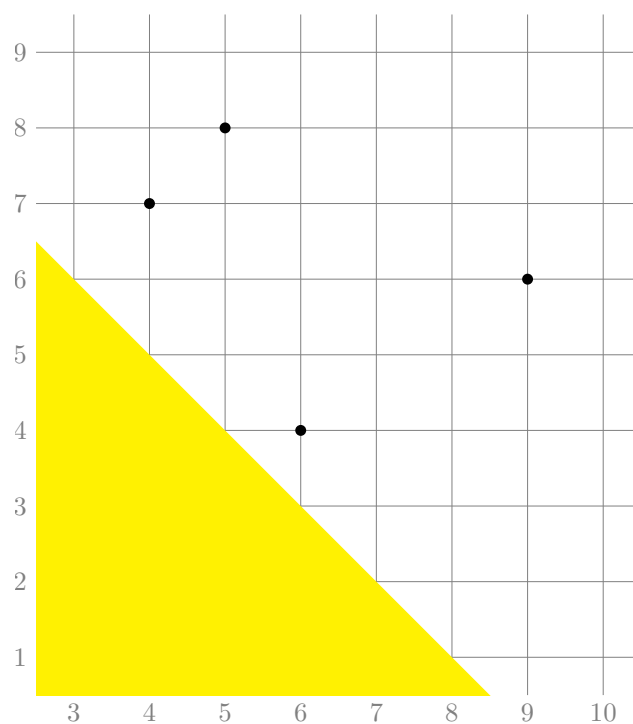
Примеры

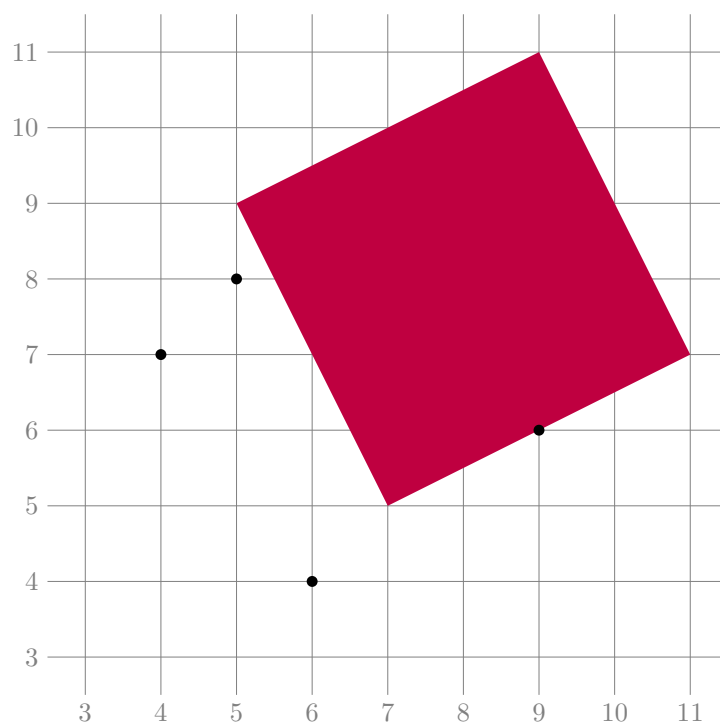
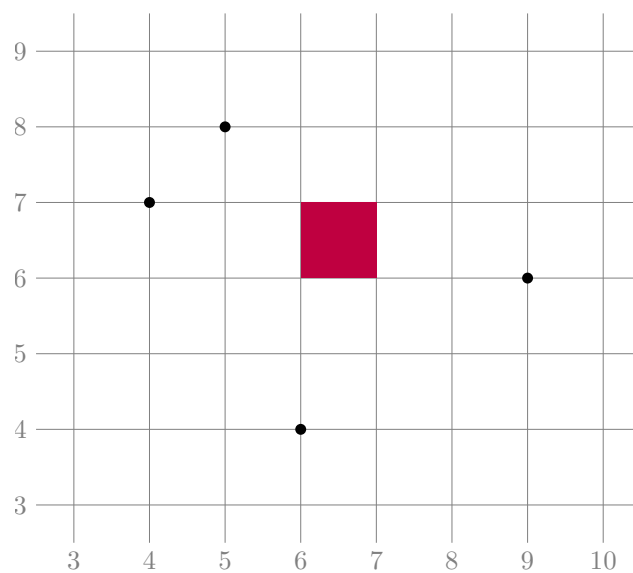
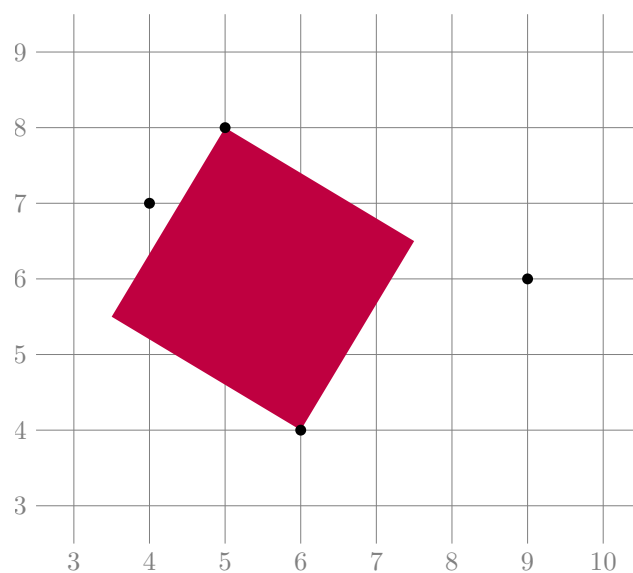
стандартный ввод	стандартный вывод
4 4	Yes
4 7	Yes
5 8	No
6 4	No
9 6	
1 9 11 8 4	
1 3 7 8 2	
1 3 6 8 1	
1 13 6 3 11	
4 8	Yes
4 7	Yes
5 8	No
6 4	No
9 6	Yes
1 9 11 8 4	No
1 3 7 8 2	Yes
1 3 6 8 1	Yes
1 13 6 3 11	
2 6 4 5 8	
2 6 6 7 7	
2 7 5 9 11	
2 5 3 6 6	

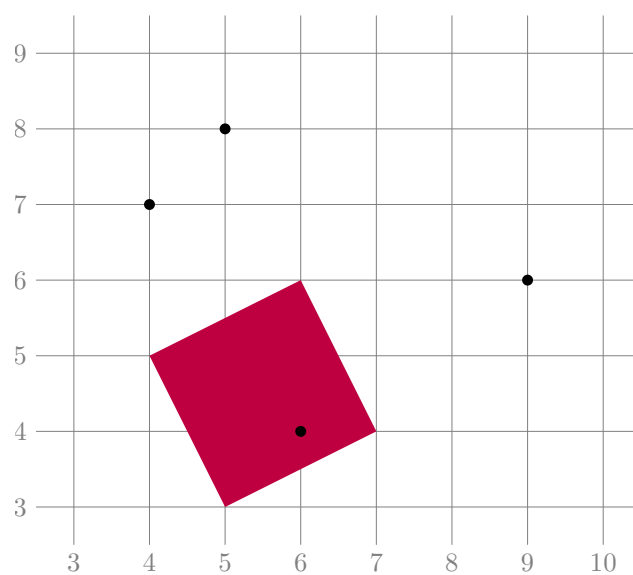
Замечание

Ниже приведены точки и области для каждого из случаев второго примера









Задача К. 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 L. 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 М. Полёт над озером

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

Петя любит делать снимки местности при помощи квадрокоптера. Однажды он отправился в путешествие к живописным берегам озера. Озеро может быть представлено в виде **не обязательно выпуклого** многоугольника на плоскости. Петя запустил квадрокоптер над озером на небольшом расстоянии от поверхности воды и сделал несколько снимков. Если квадрокоптер находится над точкой озера с координатами (x', y') , то на снимке будут видны все точки плоскости, y -координаты которых не превосходят y' . Для каждого снимка Петю интересует суммарная площадь частей озера, которые видны на этом снимке. Но он очень занят обработкой фотографий и просит Вас помочь с этой задачей.

Формат входных данных

В первой строке задано целое число n ($3 \leq n \leq 100\,000$) — количество вершин многоугольника, описывающего озеро.

Каждая из следующих n строк содержит по два целых числа x_i, y_i ($0 \leq x_i, y_i \leq 10^9$) — координаты вершин многоугольника в порядке обхода.

Гарантируется, что озеро представляет собой корректный невырожденный многоугольник, то есть не содержит самопересечений и самокасаний, также никакие три подряд идущие вершины многоугольника не лежат на одной прямой. Обратите внимание, что многоугольник может быть **невыпуклым**.

Следующая строка содержит целое число q ($1 \leq q \leq 100\,000$) — количество снимков.

Каждая из следующих q строк содержит по одному целому числу y'_i ($0 \leq y'_i \leq 10^9$) — y -координату точки озера, над которой находился квадрокоптер, чтобы сделать i -й снимок.

Формат выходных данных

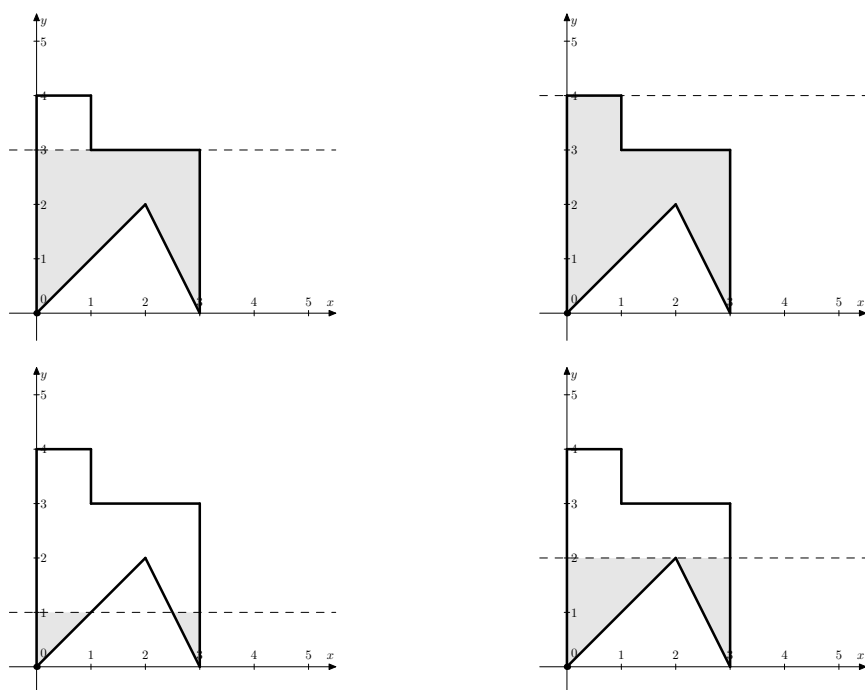
Выведите q чисел s_i — суммарную площадь озера, которую видно на i -й фотографии. Можно доказать, что ответ всегда представим в виде несократимой дроби $\frac{P}{Q}$. Вам нужно вывести $s_i \equiv P \cdot Q^{-1} \pmod{10^9 + 7}$. Иными словами, вам необходимо найти такое s_i ($0 \leq s_i < 10^9 + 7$), что $s_i \cdot Q \equiv P \pmod{10^9 + 7}$.

Примеры

стандартный ввод	стандартный вывод
7 3 0 3 3 1 3 1 4 0 4 0 0 2 2 4 1 3 2 4	750000006 6 3 7
11 5 7 8 7 9 5 12 7 14 7 14 5 12 5 12 2 1 2 1 9 3 9 10 5 8 6 3 13 7 2 1 9 4	33 500000062 45 11 61 55 0 0 61 22

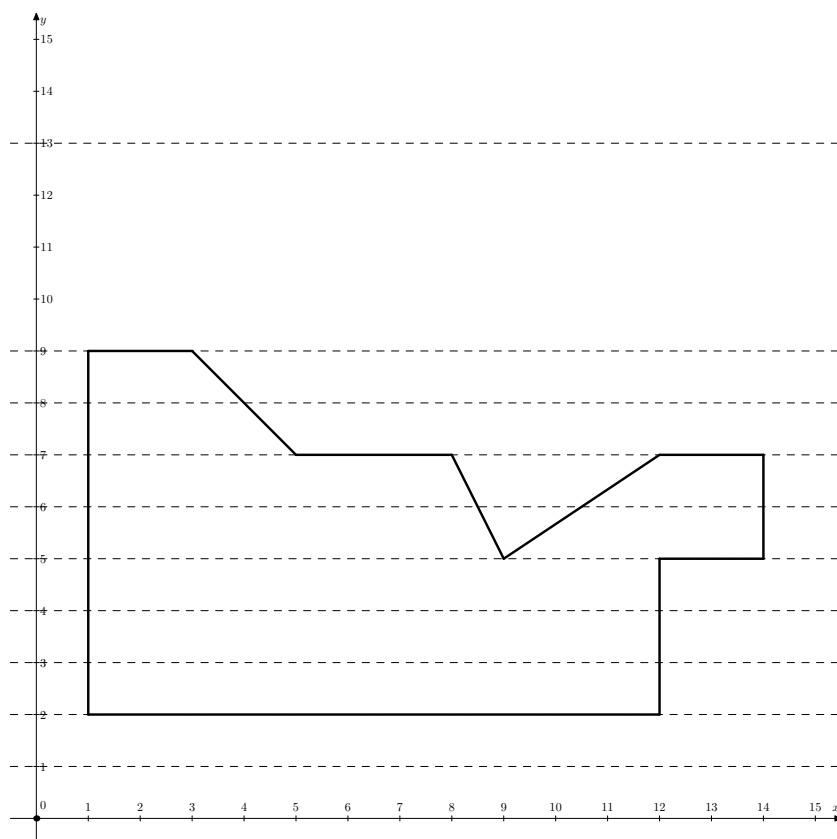
Замечание

В первом тесте из условия озеро и области снимков выглядят так:



На первой фотографии площадь видимой части озера равна $\frac{3}{4}$, на остальных снимках площадь видимой части озера является целым числом.

Во втором тесте из условия озеро и области снимков выглядят так:



На второй фотографии площадь видимой части озера равна $58\frac{1}{2} = \frac{117}{2}$, на остальных снимках площадь видимой части озера является целым числом.

Задача N. 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