Задача A. Выпуклая оболочка

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

Дано N точек на плоскости.

Нужно построить их выпуклую оболочку.

Гарантируется, что выпуклая оболочка не вырождена.

Формат входных данных

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

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

Формат выходных данных

В первой строке выведите N число вершин выпуклой оболочки. Следующие N строк должны содержать координаты вершин в порядке обхода. Никакие три подряд идущие точки не должны лежать на одной прямой. Кроме того, в последней строке выведите площадь получившейся выпуклой оболочки. Площадь необходимо вывести абсолютно точно.

Примеры

hull.in	hull.out
5 0 0 2 0 0 2 1 1 2 2	4 0 0 0 2 2 2 2 0 4.0

Задача B. Платные дороги

Имя входного файла:	`highways.in`
Имя выходного файла:	`highways.out`
Ограничение по времени:	3 секунды
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 — дальнобойщики 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 ($1 \leqslant n, m \leqslant 100\,000$).

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

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) ($|x|, |y| \leqslant 10^9$).

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

highways.in	highways.out
4 2 0 1 0 1 0 1 1 1 0 1 1 -1 0 0 2 0	3 1 3 4

Задача C. Формула-3

Имя входного файла:	`formula.in`
Имя выходного файла:	`formula.out`
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

В ежегодном чемпионате Флатландии (которая, естественно, является плоским миром) по космическим гонкам «Формула-3» участвуют N космических скутеров, имеющие форму треугольников. До начала гонок скутеры занимают положение в стартовой зоне согласно результатам жеребьевки.

Скутеры стартуют строго по порядку. Каждый скутер, получив команду «старт», уезжает в положительном направлении оси Ox. Следующий скутер стартует лишь тогда, когда предыдущий покинет стартовую зону. Скутеры уезжают строго параллельно оси Ox, скутеры в стартовой зоне не поворачивают и не разворачиваются.

Естественно, что если в момент старта на пути скутера окажется другой скутер, то произойдет авария (даже если скутер заденет лишь угол другого скутера своим углом).

Для уменьшения опасности столкновения скутеров на старте строго соблюдается следующее правило: прямые, параллельные оси Ox и пересекающие какой-то скутер, должны в совокупности пересекать не более 100 других скутеров (прямая, проходящая через одну точку скутера также считается прямой, пересекающей скутер). Например, на приведенном рисунке прямые, параллельные Ox и пересекающие скутер 2, проходят через 2 других скутера (1 и 3), а прямые, проходящие через скутер 1, проходят только через один другой скутер (номер 2).

Главный Судья гонок хочет определить порядок, в котором должны стартовать скутеры, чтобы аварии не произошло. Например, в ситуации, приведенной на рисунке, сначала должен стартовать скутер номер 2 (если попытается стартовать скутер номер 1 или 3, то он столкнется со скутером номер 2). После этого скутеры 1 и 3 могут стартовать в любом порядке (они друг другу не мешают).

Помогите Главному Судье — напишите программу, которая определит какой-нибудь порядок старта скутеров, чтобы аварии не произошло.

Формат входных данных

В первой строке входного файла записано натуральное число N ($1 \leqslant N \leqslant 30\,000$).

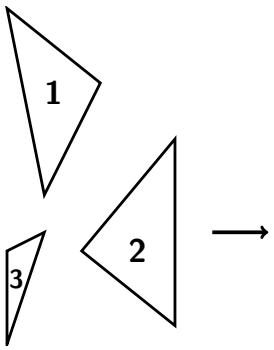
В каждой из следующих N строк записано по 6 чисел: x_1, y_1, x_2, y_2, x_3, y_3 — координаты трех вершин скутера на старте, целые числа, не превосходящие по модулю 10^6. В начальный момент скутеры не задевают друг друга.

Формат выходных данных

В единственной строке выходного файла выведите перестановку чисел от 1 до n — порядок старта скутеров, при котором не произойдёт авария. Если ответов несколько, разрешается вывести любой из них.

Примеры

formula.in	formula.out
3 1 19 3 9 6 15 5 6 10 2 10 12 1 1 6 1 3 7	2 1 3
3 0 1 -2 1 -1 -1 5 6 10 2 10 12 1 1 6 1 3 7	2 3 1


1
2
3

Задача D. Морской бой Light

Имя входного файла:	`battleship.in`
Имя выходного файла:	`battleship.out`
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

N вражеских кораблей заняли свои позиции и готовятся к атаке. Вакуумная бомба уничтожает все объекты в радиусе R от точки взрыва (то есть все объекты, расстояние от которых до точки взрыва не больше R).

Требуется определить, за какое наименьшее количество взрывов можно уничтожить все корабли.

Формат входных данных

В первой строке входных данных задаются целые числа N ($2 \leqslant N \leqslant 10$) и R ($0 < R \leqslant 15\,000$). В следующих N строках содержится по два целых числа — координаты короблей, по модулю не превосходящие 10^5.

Формат выходных данных

Выведите единственное число — необходимое количество взрывов

Примеры

battleship.in	battleship.out
10 1 0 0 0 1 0 2 0 3 0 4 0 5 0 6 0 7 0 8 0 9	4

Задача E. Ещё один спектакль в ЛКШ

Имя входного файла:	`ambitious.in`
Имя выходного файла:	`ambitious.out`
Ограничение по времени:	5 секунд
Ограничение по памяти:	512 мегабайт

Всемирно известный ЛКШ-атский классик Олег ставит очередной спектакль. В самой эпичной сцене спектакля будут участвовать n преподавателей, m культоргов и t админов. По его задумке, выбираются один преподаватель, один культорг и два админа. Между преподавателем и культорогом протягивается гирлянда красного цвета. Если она пересекает гирлянду зеленого цвета, протянутую между двумя админами, то над сценой возникает небольшой фейерверк синего цвета. Для того, чтобы сцены выглядела как можно более зрелищно, Олег хочет максимизировать количество фейерверков. Он просит вас помочь ему и посчитать количество возможных фейерверков.

Представим преподавателей, культоргов и админов точками на плоскости. При этом, y-координата всех преподавателей больше 0, а y-координата всех культоргов и админов меньше 0. Никакие две точки не совпадают, никакие три точки не лежат на одной прямой. От вас требуется посчитать количество неупорядоченных четверок $\{P, K, A_1, A_2\}$ таких, что отрезок P — преподаватель, K — культорг, A_1 и A_2 — админы, и отрезок PK пересекает отрезок A_1A_2.

Формат входных данных

Входной файл содержит несколько тестов.

Каждый тест начинается с целого числа n — количество преподавателей ($1 \leqslant n \leqslant 1500$). За ним следует n строк, которые содержат целые числа px_i, py_i — координаты i-го преподавателя. Далее записано число m — количество культоргов ($1 \leqslant m \leqslant 1500$). В следующих m строках содержатся по два целых числа kx_i, ky_i — координаты i-го культорга. После этого следует число t — количество админов ($1 \leqslant t \leqslant 1500$). Последующие t строк содержат целые числа ax_i, ay_i — координаты i-го админа.

Координаты всех преподавателей удовлетворяют неравенствам $-10^9 \leqslant px_i \leqslant 10^9$, $0 < py_i \leqslant 10^9$. Координаты всех культоргов и админов удовлетворяют неравенствам $-10^9 \leqslant kx_i, ax_i \leqslant 10^9$, $-10^9 \leqslant ky_i, ay_i < 0$.

Общее количество преподавателей во всех тестах не превосходит 1500. Общее количество культоргов во всех тестах не превосходит 1500. Общее количество админов во всех тестах не превосходит 1500.

Формат выходных данных

Для каждого теста выведите одно целое число: количество возможных фейерверков.

Примеры

ambitious.in	ambitious.out
3 1 12 10 30 30 10 1 10 -10 4 2 -11 9 -1 11 -1 15 -14 0	7

Задача F. Силовое поле

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

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 ($3 \leqslant N \leqslant 1\,000$) и ограничение на абсолютную величину их координат ($5 \leqslant C \leqslant 1\,000\,000$).

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.

Примеры

ввод	вывод
4 5 0 4 0 3 0 3 0 2 1 1 3 1 3 0 3 0	? -1 3 1 3 ? -1 2 1 2 ? -1 1 0 2 ? -1 0 0 2 ? 0 0 0 2 ? 1 0 1 2 ? 2 0 2 2 ? 3 0 1 2 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».