Highways. Платные дороги

Имя входного файла: `highways.in`
Имя выходного файла: `highways.out`

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 входного файла

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 $ax + by + c = 0$ ($|a|, |b|, |c| \leqslant 10^6$).

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) ($|x|, |y| \leqslant 10^6$).

Формат выходного файла

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

Hull. Выпуклая оболочка

Имя входного файла: `hull.in`
Имя выходного файла: `hull.out`

Дано n точек на плоскости. Постройте их выпуклую оболочку. Гарантируется, что выпуклая оболочка не вырождена.

Формат входного файла

В первой строке записано число n ($3 \leqslant n \leqslant 10^5$). Следующие n строк содержат пары целых чисел x и y ($-10^9 \leqslant x, y \leqslant 10^9$). Будьте аккуратны! Точки произвольны. Бывают совпадающие, бывают лежащие на одной прямой в большом количестве.

Формат выходного файла

В первой строке выведите k — число вершин выпуклой оболочки. Следующие k строк должны содержать координаты вершин в порядке обхода. Никакие три подряд идущие точки не должны лежать на одной прямой.

Пример

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

Diameter. Диаметр множества точек

Имя входного файла: `diameter.in`
Имя выходного файла: `diameter.out`

Дано n точек с целочисленными координатами. Найдите квадрат максимальной длины отрезка с концам в двух различных точках данного множества.

Формат входного файла

В первой строке дано целое число n ($1 \leqslant n \leqslant 10^5$). Следующие n строк содержат пары целых чисел x и y ($-10^9 \leqslant x, y \leqslant 10^9$). Будьте аккуратны! Точки произвольны. Бывают совпадающие, бывают лежащие на одной прямой в большом количестве.

Формат выходного файла

Выведите квадрат максимальной длины отрезка с концам в двух различных точках данного множества.

Примеры

diameter.in	diameter.out
1 3 4	0
2 2 0 0 2	8
4 1 1 4 1 1 5 1000 1000	1996002

Closest. Минимальный периметр

Имя входного файла: `closest.in`
Имя выходного файла: `closest.out`

Дано n точек с целочисленными координатами. Найдите минимальный периметр треугольника с вершинами в трёх различных точках данного множества (треугольник может быть вырожденным).

Формат входного файла

Первая строка содержит целое число n ($1 \leqslant n \leqslant 100\,000$) — количество точек. Следующие n строк содержат координаты точек x_i и y_i ($0 \leqslant x_i, y_i \leqslant 10^9$). Все точки различные.

Формат выходного файла

Выведите минимальный периметр. Правильным считается ответ если его относительная или абсолютная погрешность не превосходит 10^{-9}.

Примеры

closest.in	closest.out
4 1 1 4 1 1 5 1000 1000	12
9 100000 200000 200000 200000 150000 286603 60000 140000 240000 140000 150000 340000 1 340000 300000 340000 150000 87087	300000.7960884156