

Задача А. Платные дороги

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
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 ($1 \leq n, m \leq 100\,000$).

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

stdin	stdout
4 2 0 1 0 1 0 1 1 1 0 1 1 -1 0 0 2 0	3 1 3 4

Задача В. Принцесса

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

Принцесса Евлампия живет в замке, окруженном забором. Жизнь принцессы тяжела, но при этом и очень интересна. Главным ее развлечением является общение с многочисленными поклонниками, постоянно прибывающими из соседних замков, городов и даже королевств.

Замок принцессы окружен забором, представляющим из себя выпуклый многоугольник. Отец принцессы, король, достаточно строг, поэтому всем поклонникам принцессы приходится попадать туда через единственную во всем заборе дырку, вместо того, чтобы войти на территорию замка через парадные ворота. Дырка находится в одной из вершин многоугольника. При этом, если пройти напрямую к дырке поклоннику не удастся, ему приходится обходить забор вдоль его периметра. Естественно, каждому поклоннику интересно, сколько ему придется пройти, чтобы попасть из точки своего начального местоположения к дырке, и все спрашивают об этом принцессу, перед тем как прийти к ней в гости.

Принцесса составила список начальных местоположений всех своих поклонников и описание забора вокруг замка. Вам необходимо для каждого поклонника сообщить длину кратчайшего пути от точки его начального положения до точки, в которой находится дырка. При этом, естественно, ни одна точка этого пути не должна лежать внутри многоугольника, представляющего забор, но может лежать на его границе.

Формат входных данных

В первой строке входного файла находятся два целых числа n и k ($3 \leq n \leq 100\,000$, $1 \leq k \leq n$) — количество вершин в многоугольнике, представляющем забор, и номер вершины, в которой находится дырка. В следующих n строках содержатся пары целых чисел x_i и y_i , описывающих координаты вершин многоугольника в порядке обхода против часовой стрелки.

В следующей строке дано одно целое число m ($1 \leq m \leq 100\,000$) — количество поклонников принцессы. В следующих m строках содержатся пары целых чисел x_i и y_i , описывающих координаты начального положения очередного поклонника.

Все координаты не превышают 10^9 по абсолютной величине.

Формат выходных данных

Для каждого поклонника выведите одно число — ответ на задачу. Ответ должен отличаться от правильного не более, чем на 10^{-5} .

Примеры

<code>princess.in</code>	<code>princess.out</code>
4 2	3.23606797749979
0 1	2.0
0 0	
1 0	
1 1	
2	
2 2	
-2 0	

Задача С. Диаметр точек

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

На плоскости даны N точек. Вам требуется найти квадрат расстояния 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 квадрат расстояния между двумя наиболее удалёнными точками.

Примеры

stdin	stdout
5 0 0 2 2 1 1 0 2 2 0	8
7 0 0 1 1 2 2 0 2 1 3 0 1 2 0	10

Задача D. Идол Могоху-Ри

Имя входного файла:	<code>stdin</code>
Имя выходного файла:	<code>stdout</code>
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Давным давно где-то в глубине Америки существовало могущественное племя с великим вождем Пиннивухом. Однажды это племя захватило три города Майя. Пиннивух задумался: необходимо как-то контролировать покоренные территории. Для этого он обратился к жрецам верховного бога Могоху-Ри за помощью.

Жрецы донесли до него волю бога: для контроля этих трех городов необходимо поставить идол Могоху-Ри, создающий религиозное поле над городами. Однако этот идол настолько могущественный, что будучи неуравновешенным тремя жертвенными алтарями, расположенными по одному в каждом городе, он может запросто свести всех людей в округе с ума. Для уравнивания идола алтари нужно поставить таким образом, чтобы центр масс системы из этих трех точек совпадал с идиолом. При подсчете центра масс считается что все алтари имеют одинаковую массу.

Теперь Пиннивух размышляет: куда же поставить идол. У него есть список возвышенностей, пригодных для установки идола. Помогите ему определить, на какие из них можно поставить идол без риска выжечь мозги населению городов религиозным полем.

Каждый город имеет форму выпуклого многоугольника, никакие три вершины которого не лежат на одной прямой. Города могут пересекаться. Каждый алтарь должен прикрепляться к своему городу особым обрядом, причем он обязан находиться на его территории (возможно на границе). Таким образом, на территории города может быть несколько алтарей, но к нему будет относиться ровно один из них. Алтари, идол и возвышенности являются точками на плоскости, некоторые из которых могут совпадать.

Возвышенности рассматриваются независимо друг от друга, расположение алтарей для разных возвышенностей может быть разным.

Формат входных данных

Сначала идут разделенные пустыми строками описания трех городов в следующем формате:

В первой строке идет целое число n ($3 \leq n \leq 5 \cdot 10^4$) — количество вершин многоугольника. Следующие n строк содержат по два целых числа x_i, y_i — координаты i -ой вершины многоугольника в порядке обхода против часовой стрелки.

После описания городов идет целое число m ($1 \leq m \leq 10^5$) — количество возвышенностей. Следующие m строк содержат по два целых числа x_j, y_j — координаты j -ого холма.

Все координаты во входных данных не превосходят $5 \cdot 10^8$ по модулю.

Формат выходных данных

Выведите для каждой возвышенности на отдельной строке «YES» (без кавычек) или «NO» (без кавычек), в зависимости от того, можно ли поставить три жертвенных алтаря для уравнивания идола или нет.

Примеры

stdin	stdout
3	NO
0 0	YES
1 0	NO
1 1	YES
	NO
4	
8 8	
5 5	
6 4	
8 4	
3	
-1 -1	
-3 -1	
-2 -2	
5	
0 0	
2 1	
7 1	
1 1	
5 3	

Задача Е. Ну, погоди!

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 ($3 \leq n \leq 2000$).

Следующие n строк содержат пары целых чисел x_i, y_i — координаты точек, являющихся вершинами многоугольника ($-200\,000 \leq x_i, y_i \leq 200\,000$) 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 ($1 \leq m \leq 1000$).

Следующие m строк содержат пары целых чисел u_i, v_i — координаты точки, в которой Петя собирается посадить дерево в i -й день ($-200\,000 \leq u_i, v_i \leq 200\,000$). 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.

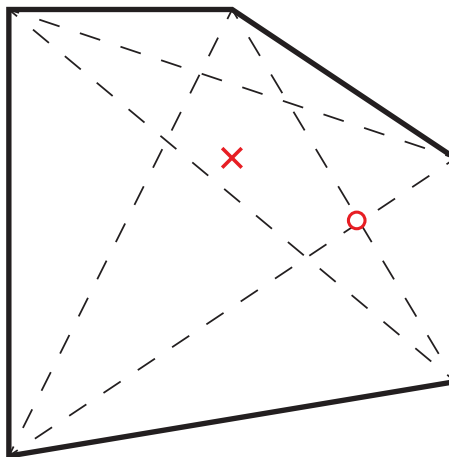
В i -й строке выходного файла выведите одно число — максимальное расстояние, на котором Вася может спрятаться от Пети, когда Петя будет сажать дерево в i -й день. Ваш ответ будет считаться правильным, если его абсолютная или относительная погрешность относительно правильного ответа составит не более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



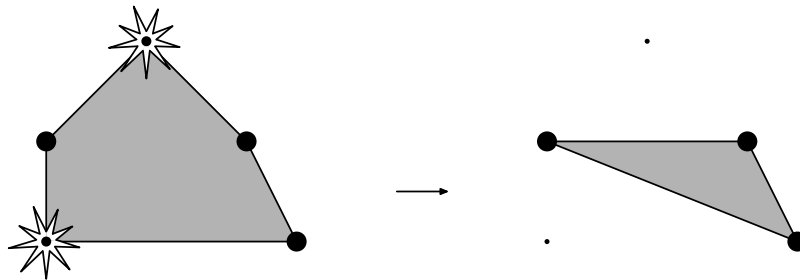
Задача F. База в джунглях

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Глубоко в джунглях расположена военная база. Она окружена n сторожевыми башнями с ультразвуковыми генераторами. В этой задаче сторожевые башни считаются точками на плоскости.

Сторожевые башни генерируют ультразвуковое поле с помощью которого защищают все объекты, расположенные строго внутри выпуклой оболочки сторожевых башен. Не существует сторожевой башни строго внутри выпуклой оболочки и никакие три сторожевые башни не находятся на одной прямой.

Враг может уничтожить несколько башен. Если такое происходит, защищаемая область уменьшается до выпуклой оболочки оставшихся башен.



Командир базы хочет построить штаб внутри защищаемой области. Чтобы максимизировать ее безопасность, он хочет максимизировать минимальное количество башен, который должен уничтожить враг, чтобы оставить штаб без защиты.

Формат входных данных

Первая строка входного файла содержит одно число n ($3 \leq n \leq 50\,000$) — количество сторожевых башен. Следующие n строк содержат декартовы координаты сторожевых башен, по одной паре в строке. Все координаты являются целыми числами и не превосходят по модулю 10^6 .

Башни описаны в порядке обхода их выпуклой оболочки по часовой стрелке.

Формат выходных данных

Выведите в выходной файл количество сторожевых башен, которые должен уничтожить враг, чтобы оставить штаб без защиты, если штаб будет расположен оптимально.

Примеры

stdin	stdout
3 0 0 50 50 60 10	1
5 0 0 0 10 10 20 20 10 25 0	2