

Задача А. 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 удалёнными 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

<code>stdin</code>	<code>stdout</code>
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 В. Место встречи изменить нельзя

Имя входного файла: `rendezvous.in`
Имя выходного файла: `rendezvous.out`
Ограничение по времени: 1 секунда
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

<code>rendezvous.in</code>	<code>rendezvous.out</code>
4	0 0
0 0	0 1
0 1	
1 1	
1 0	

Задача С. Нелётная погода

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 0.5 секунд
Ограничение по памяти: 256 мегабайт

Все вылеты из аэропорта Кольцово отложены. Главный диспетчер заявил, что, пока над территорией аэропорта висит подозрительная грозовая туча, он не позволит ни одному борту подняться в воздух. Впрочем, он заверил пассажиров, что ветер сносит тучу со скоростью один метр в секунду и скоро небо над аэропортом очистится. Правда, о направлении ветра он ничего не сказал.

Один из пассажиров нашёл в интернете спутниковый снимок территории аэропорта, сделанный ровно минуту назад. По снимку можно точно определить положение тучи на тот момент времени. Точные координаты аэропорта тоже найти нетрудно. Хватит ли этой информации, чтобы вычислить минимальное время, через которое диспетчер может дать добро на вылет самолётов?

Формат входных данных

Тучу и территорию аэропорта на снимке можно приближённо считать невырожденными строго выпуклыми многоугольниками на плоскости. В первой строке записаны целые числа n и m — количество вершин в многоугольнике, задающем территорию аэропорта, и количество вершин в многоугольнике, задающем тучу ($3 \leq n, m \leq 50\,000$). В следующих n строках записаны координаты территории аэропорта в порядке обхода против часовой стрелки. Далее в аналогичном формате задано положение тучи. Все координаты указаны в метрах и являются целыми числами, не превосходящими 10^8 по модулю. Гарантируется, что на снимке туча закрывает хотя бы одну точку внутренней области аэропорта.

Формат выходных данных

Выведите единственное число — минимальное количество секунд, которое может пройти до того момента, когда ни одна точка территории аэропорта не будет находиться под тучей. Выведите ответ с абсолютной или относительной погрешностью не более 10^{-6} . Если уже сейчас туча может не закрывать ни одну точку территории аэропорта, выведите 0.

Примеры

стандартный ввод	стандартный вывод
4 4	540.000000000000000000000000
400 400	
600 400	
600 600	
400 600	
0 0	
1000 0	
1000 1000	
0 1000	

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 уравновешивания идола алтари нужно поставить таким образом, чтобы центр масс системы из этих трех точек совпадал с идолом. 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 уравновешивания 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. Адская мухобойка

Имя входного файла: `circlecover.in`
Имя выходного файла: `circlecover.out`
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} .

Примеры

<code>circlecover.in</code>	<code>circlecover.out</code>
3	1.00 1.00
0 2	1.4142135624
0 0	
2 0	

Задача F. Разделите точки

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

На плоскости нарисованы точки черного и белого цветов. Можно ли провести такую прямую, что она не проходит ни через одну из этих точек, и при этом все точки черного цвета лежат в одной её полуплоскости, а все точки белого цвета — в другой?

Формат входных данных

В первой строке через пробел заданы два целых числа n и m — количество точек черного и белого цветов ($1 \leq n, m \leq 100$). В следующих $n + m$ строках через пробел записаны пары целых чисел x_i и y_i — координаты точек ($0 \leq x_i, y_i \leq 10\,000$). Первые n строк задают точки черного цвета, следующие m строк — белого.

Гарантируется, что никакие две точки не совпадают.

Формат выходных данных

Выведите «YES», если искомая прямая существует, и «NO» — в противном случае.

Примеры

стандартный ввод	стандартный вывод
3 3 100 700 200 200 600 600 500 100 500 300 800 500	YES
3 3 100 300 400 600 400 100 600 400 500 900 300 300	NO

Замечание

Решайте эту задачу как будто точек много. Все решения за квадрат и выше будут отклонены.

Задача G. Растягивание плоскости

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	10 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 α_j раз по x -координате.

Более формально, у Игоря есть q запросов, в j -м из которых для числа α_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 α_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

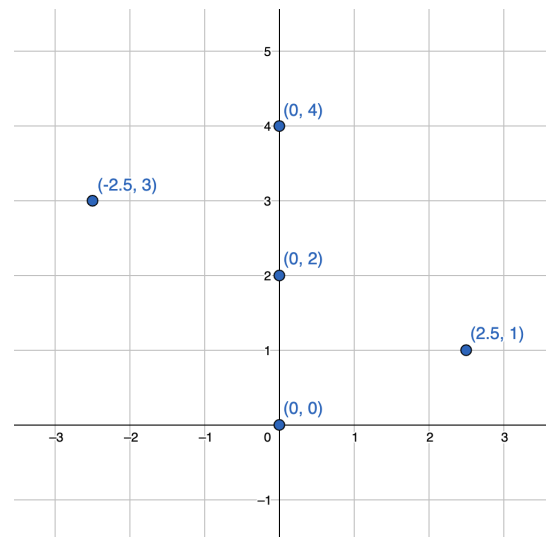
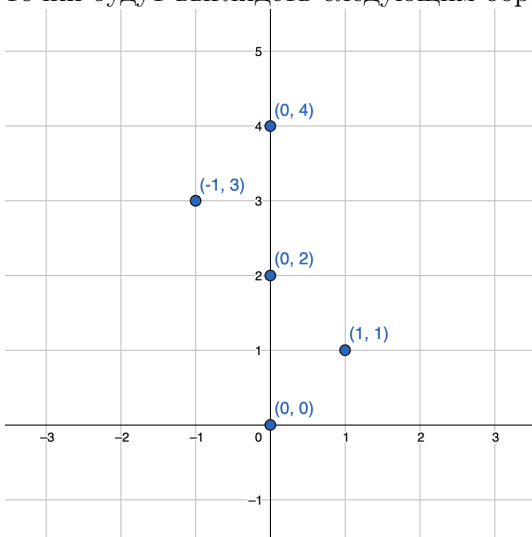
Для каждого набора входных данных выведите q строк, в i -й строке должно содержаться единственное вещественное число — ответ на i -й запрос. Ответ будет считаться правильным, если его абсолютная или относительная погрешность не превышает 10^{-6} . Более формально, если a — ваш ответ, а b — ответ жюри, 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 $(-10.5, 13)$.

Задача Н. Дом у дороги

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

Министерство дорожного транспорта решило построить себе новый офис. Поскольку министр регулярно выезжает с инспекцией наиболее важных трасс, было решено, что офис министерства не должен располагаться слишком далеко от них.

Наиболее важные трассы представляют собой прямые на плоскости. Министерство хочет выбрать такое расположение для своего офиса, чтобы максимум из расстояний от офиса до трасс был как можно меньше. Требуется написать программу, которая по заданному расположению наиболее важных трасс определяет оптимальное расположение дома для офиса министерства дорожного транспорта.

Формат входных данных

Первая строка входного файла содержит одно целое число n — количество наиболее важных трасс ($1 \leq n \leq 10^4$).

Последующие n строк описывают трассы. Каждая трасса описывается четырьмя целыми числами x_1, y_1, x_2 и y_2 и представляет собой прямую, проходящую через точки (x_1, y_1) и (x_2, y_2) . Координаты заданных точек не превышают по модулю 10^4 . Точки (x_1, y_1) и (x_2, y_2) ни для какой прямой не совпадают.

Формат выходных данных

Выходной файл должен содержать два разделенных пробелом вещественных числа: координаты точки, в которой следует построить офис министерства дорожного транспорта. Координаты по модулю не должны превышать 10^9 , гарантируется, что хотя бы один такой ответ существует. Если оптимальных ответов несколько, необходимо вывести любой из них.

Ответ должен иметь абсолютную или относительную погрешность не более 10^{-6} , что означает следующее. Пусть максимальное расстояние от выведенной точки до некоторой трассы равно x , а в правильном ответе оно равно y . Ответ будет засчитан, если значение выражения $\frac{|x-y|}{\max(1,|y|)}$ не превышает 10^{-6} .

Пример

house.in	house.out
4 0 0 0 1 0 0 1 0 1 1 2 1 1 1 1 2	0.5000000004656613 0.4999999995343387