

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 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

<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. Новый почтовый клиент

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	4 секунды
Ограничение по памяти:	256 мегабайт

Вы работаете в крупной компании по изучению общественного мнения, и вас попросили срочно предоставить данные касательно мнения населения о новом государственном почтовом клиенте. У вас есть результаты опроса n граждан, для i -го из которых у вас есть следующая информация:

- a_i — количество цифр числа π , которое помнит респондент
- b_i — количество волос на его голове
- c_i — нравится ли ему новый почтовый клиент

Однако, посмотрев на данные, вы не увидели корреляции между a , b и c , и начали подозревать, что надо было задавать респондентам другие вопросы.

Как опытный аналитик, вы подумали, что, возможно, данные будут более осмысленными, если взять какие-нибудь вещественные числа S и T , и отсортировать тройки (a_i, b_i, c_i) по величине $a_i \cdot S + b_i \cdot T$. Данные будут выглядеть тем лучше, чем плотнее сгруппированы те тройки, в которых c_i истинно. Более формально, пусть j и k — индексы первой и последней тройки, в которых c_i истинно. Тогда вы хотите минимизировать *размер группы*, определяемый как $k - j + 1$. Заметим, что некоторые тройки могут иметь равное значение $a_i \cdot S + b_i \cdot T$, в этом случае вы должны предполагать наихудший возможный порядок сортировки (т.е. тот, который дает максимальный размер группы для данных (S, T)).

Формат входных данных

В первой строке содержится целое число n ($1 \leq n \leq 250\,000$) — количество опрошенных граждан. Дальше идут n строк, i -я из которых содержит целые числа a_i ($0 \leq a_i \leq 2\,000\,000$), b_i ($0 \leq b_i \leq 2\,000\,000$) и c_i , где c_i равно 1, если i -му опрошенному нравится новый клиент, и 0 иначе. Гарантируется, что есть хотя бы один человек, которому нравится новый клиент.

Формат выходных данных

Выведите наименьший возможный размер группы среди всех возможных пар (S, T) .

Примеры

стандартный ввод	стандартный вывод
6 0 10 0 10 0 1 12 8 1 5 5 0 11 2 1 11 3 0	4
10 6 1 1 0 2 0 2 1 1 6 1 1 8 2 0 4 4 0 4 0 0 2 3 1 6 1 0 6 3 1	8
5 5 7 0 3 4 0 5 7 0 5 7 1 9 4 0	1