

Содержание

Локализация точки	2
Задача А. Внутренняя точка [0.5 sec, 256 mb]	2
Выпуклые оболочки	3
Задача В. Выпуклая оболочка [0.5 sec, 256 mb]	3
Задача С. Стена [0.5 sec, 256 mb]	4
Задача D. Армейская история [0.25 sec, 256 mb]	5
Задача Е. Селфи на плоскости [0.5 sec, 256 mb]	6
Заметающая прямая	8
Задача F1. Звёзды [0.2 sec, 256 mb]	8
Задача F2. Окна [1.0 sec, 256 mb]	9
Задача F3. Золотые рудники [0.3 sec, 256 mb]	10
Задача F4. Внутренняя точка 2 [3.5 sec, 256 mb]	11
Задача F5. Антиуфология [1.0 sec, 256 mb]	12
Деревянные структуры	13
Задача G1. Отрезки, пересекающие прямоугольник [2 sec, 256 mb]	13
Задача G2. Отрезки, пересекающие прямоугольник [6 sec, 256 mb]	14
Герман и четырёхмерная модель теории относительности	15
Задача H1. Не лётная погода [0.3 sec, 256 mb]	15
Задача H2. Лоцман [0.5 sec, 256 mb]	16
3D	17
Задача I. The Worm in the Apple [0.5 sec, 256 mb]	17

В некоторых задачах большой ввод и вывод. Пользуйтесь **быстрым вводом-выводом**.

Обратите внимание на компилятор GNU C++11 5.1.0 (TDM-GCC-64) inc, который позволяет пользоваться **дополнительной библиотекой**. Под ним можно сдать **вот это**.

Локализация точки

Задача А. Внутренняя точка [0.5 sec, 256 mb]

Дан строго выпуклый N -угольник и K точек. Для каждой точки нужно определить, где она находится — внутри, на границе, или снаружи.

Формат входных данных

N ($3 \leq N \leq 10^5$). Далее N точек — вершины многоугольника.

K ($0 \leq K \leq 10^5$). Далее K точек — запросы.

Все координаты — целые числа по модулю не превосходящие 10^7 .

Формат выходных данных

Для каждого запроса одна строка — INSIDE, BORDER или OUTSIDE.

Примеры

stdin	stdout
4	INSIDE
0 0	BORDER
2 0	BORDER
2 2	OUTSIDE
0 2	
4	
1 1	
0 0	
0 1	
0 3	

Выпуклые оболочки

Задача В. Выпуклая оболочка [0.5 sec, 256 mb]

Дано N точек на плоскости.

Нужно построить их выпуклую оболочку.

Гарантируется, что выпуклая оболочка не вырождена.

Формат входных данных

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

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

Формат выходных данных

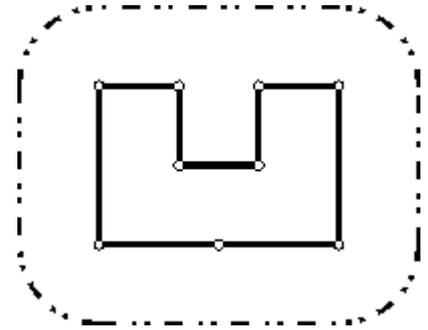
В первой строке выведите N — число вершин выпуклой оболочки. Следующие N строк должны содержать координаты вершин в порядке обхода. Никакие три подряд идущие точки не должны лежать на одной прямой.

Пример

stdin	stdout
5	4
0 0	0 0
2 0	2 0
0 2	2 2
1 1	0 2
2 2	

Задача С. Стена [0.5 sec, 256 mb]

Жил-был жадный король, который однажды приказал главному архитектору окружить королевский замок стенами. Король был настолько жаден, что не желал слушать рассказы архитектора о красивой кирпичной стене с прекрасным силуэтом и изящными высокими башнями. Вместо этого он приказал окружить замок стенами, затратив минимальное количество камня и времени, но потребовал, чтобы стена не подходила к замку ближе, чем на заданное расстояние. Если король узнает, что архитектор потратил не минимально возможное количество ресурсов, то архитектор лишится головы. Более того, король потребовал, чтобы архитектор сразу же предложил проект стены с указанием минимального количества ресурсов, необходимых для постройки.



Вы должны помочь архитектору сохранить голову, написав программу для поиска минимальной длины стены, удовлетворяющей условиям короля.

Задачу упрощает то, что замок короля имеет форму многоугольника и расположен на равнине. Архитектор уже ввел систему координат и точно измерил координаты вершин замка в футах.

Формат входных данных

Первая строка входного файла содержит числа N и L , разделенные пробелом. N ($3 \leq N \leq 1000$) — это количество вершин в королевском замке, а L ($1 \leq L \leq 1000$) — минимальное количество футов, на которое стена может приближаться к замку.

Следующие N строк описывают координаты замка в порядке обхода по часовой стрелке. В каждой строке через пробел записаны целые числа x_i и y_i , разделенные пробелом ($-10\,000 \leq x_i, y_i \leq 10\,000$), которые обозначают координаты i -ой вершины. Все вершины различны, и никакие две стороны не пересекаются кроме как по вершинам.

Формат выходных данных

Выведите минимальную длину стены в футах, удовлетворяющей условиям короля с точностью не менее 6 знаков после запятой.

Примеры

stdin	stdout
9 100	1628.3185307180
200 400	
300 400	
300 300	
400 300	
400 400	
500 400	
500 200	
350 200	
200 200	

Задача D. Армейская история [0.25 sec, 256 mb]

Военные захотели усилить охрану космодрома. Было решено, что он должен быть огорожен как можно большим числом заборов, а вдоль каждого из них вокруг космодрома должна ходить вооруженная охрана. Издали приказ, и начали готовить проект.

Желая выслужиться, прапорщик Дуров направил роту солдат вкапывать столбы для забора, не дожидаясь проекта. Солдаты, не долго думая, наставили столбов куда попало. Помогите прапорщику решить, как на этих столбах сделать побольше заборов из колючей проволоки.

Формат входных данных

В первой строке содержится целое число $3 \leq N \leq 4000$ – количество вкопанных столбов. В каждой из последующих N строк содержатся два целых числа $0 \leq x, y \leq 10\,000$ – координаты столба. Естественно, все столбы вкопаны в разных точках.

Формат выходных данных

Выведите максимально возможное количество заборов, которые можно построить. Заборы должны являться многоугольниками без самопересечений, с углами в точках столбов. Заборы не должны касаться или пересекаться, и должны быть вложенными один в другой, чтобы вдоль каждого забора патруль мог обойти космодром и снаружи, и изнутри.

Примеры

stdin	stdout
4 100 100 200 100 100 200 300 300	1

Задача Е. Селфи на плоскости [0.5 сек, 256 mb]

Для парада параллелей несколько школьников решило сделать групповое селфи. Если вдруг кто-то не знает, селфи — это такой модный вид фотографии, когда человек сам снимает себя и своих товарищей, как правило, на мобильный телефон. Школьники уже выбрали места, где будут стоять во время съёмки, и теперь пытаются определить, чьим телефоном будет сделана фотография.

В рамках данной задачи будем считать каждого школьника точкой на плоскости. Ребята решили, что фотографию стоит делать из точки, в которой находится какой-то школьник, причём угол обзора, в который помещаются все остальные точки, соответствующие школьникам, должен быть как можно меньше. Помогите им определить минимальный угол обзора на фотографии, вмещающей всех людей.

Более формально, ответ на задачу — такое минимальное неотрицательное вещественное число градусов α , что существует некоторый человек и угол величиной в α градусов с вершиной в позиции этого человека, содержащий всех людей (при этом человек может находиться на границе угла).

Длиной палки для селфи можно пренебречь.

Формат входных данных

В первой строке ввода задано единственное число n ($1 \leq n \leq 200\,000$) — количество людей.

Далее следуют n строк, каждая из которых содержит по два числа x_i и y_i ($-10^9 \leq x_i, y_i \leq 10^9$) — координаты позиции i -го человека. Некоторым людям может нравиться одно и то же место, поэтому позиции некоторых людей могут совпадать.

Формат выходных данных

Выведите единственное вещественное число α — минимальный угол обзора в градусах.

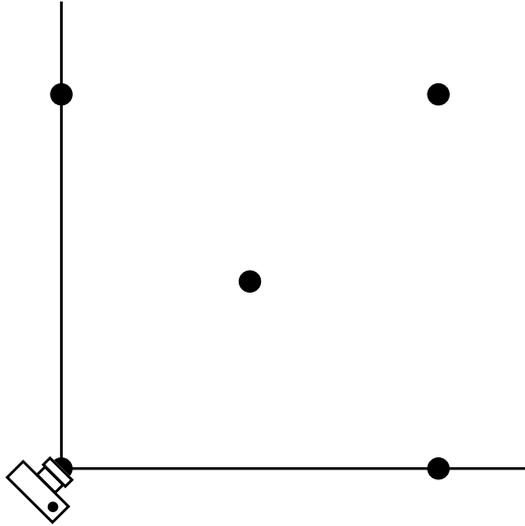
Ваш ответ будет считаться правильным, если его абсолютная или относительная погрешность относительно ответа жюри не превосходит 10^{-6} . Более формально, если ваш ответ — ans , а ответ жюри — $jury$, ваш ответ будет признан правильным, если $\frac{|ans-jury|}{\max(1, jury)} \leq 10^{-6}$.

Примеры

stdin	stdout
5 1 1 3 1 2 2 1 3 3 3	90.0000000000000000
3 1 0 2 1 3 2	0.0000000000000000
3 0 0 3 0 3 4	36.86989764584402000

Замечание

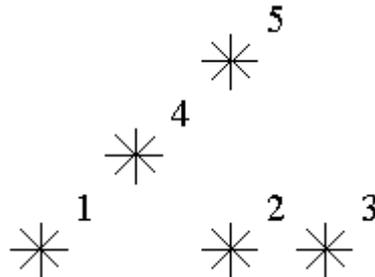
Иллюстрация для первого примера:



Заметающая прямая

Задача F1. Звёзды [0.2 sec, 256 mb]

Астрономы часто исследуют звёздные карты, на которых звёзды представлены точками на плоскости, каждая звезда имеет декартовы координаты. Пусть уровень звезды – количество звёзд, которые не выше и не правее данной звезды. Астрономы хотят найти распределение уровней звёзд.



Для примера посмотрим на карту звёзд на картинке выше. Уровень звезды номер 5 равен 3 (т.к. есть звёзды с номерами 1, 2, 4). Уровни звёзд 2 и 4 равны 1. На данной карте есть только одна звезда на уровне 0, две звезды на уровне 1, одна звезда на уровне 2 и одна звезда на уровне 3. Напишите программу, считающую количество звёзд на каждом уровне.

Формат входных данных

Вам дан один или несколько тестов. Каждый тест описывается следующим образом.

В первой строке количество звёзд N ($1 \leq N \leq 15\,000$). Следующие N строк описывают координаты звёзд (два целых числа X и Y , разделённые пробелом, $0 \leq X, Y \leq 32\,000$). В каждой точке плоскости находится не более одной звезды. Звёзды перечислены в порядке возрастания Y координаты, при равенстве в порядке возрастания X координаты.

Формат выходных данных

Выведите ответ для каждого теста. Ответ для теста описывается следующим образом. N строк, по одному числу в строке. i -я строка содержит количество звёзд на уровне i ($i = 0 \dots N-1$).

Примеры

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

Замечание

Простейший scanline.

Задача F2. Окна [1.0 сек, 256 mb]

На экране расположены прямоугольные окна, каким-то образом перекрывающиеся (со сторонами, параллельными осям координат). Вам необходимо найти точку, которая покрыта наибольшим числом из них.

Формат входных данных

В первой строке входного файла записано число окон n ($1 \leq n \leq 50\,000$).

Следующие n строк содержат координаты окон $x_{(1,i)} y_{(1,i)} x_{(2,i)} y_{(2,i)}$, где $\langle x_{(1,i)}, y_{(1,i)} \rangle$ — координаты левого верхнего угла i -го окна, а $\langle x_{(2,i)}, y_{(2,i)} \rangle$ — правого нижнего (на экране компьютера y растёт сверху вниз, а x — слева направо).

Все координаты — целые числа, по модулю не превосходящие $2 \cdot 10^5$.

Формат выходных данных

В первой строке выходного файла выведите максимальное число окон, покрывающих какую-либо из точек в данной конфигурации. Во второй строке выведите два целых числа, разделенные пробелом — координаты точки, покрытой максимальным числом окон. Окна считаются замкнутыми, т.е. покрывающими свои граничные точки.

Пример

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

Задача F3. Золотые рудники [0.3 сек, 256 mb]

Байтмен, один из заслуженных работников компании по добыче золота в Байтленде, собирается в этом году на пенсию. Начальство компании решило вознаградить его за заслуги перед отечеством. Байтмену разрешили присвоить себе прямоугольную часть земли, со сторонами s и w , параллельными осям координат, со всеми входящими туда рудниками. Положение (сдвиг) участка он выбирает сам. Назовем стоимостью участка количество рудников, лежащих внутри него и на его границе. Ваша задача вычислить максимально возможную стоимость такого участка.

Формат входных данных

В первой строке входного файла находятся два целых числа s и w , разделенных одиночным пробелом ($1 \leq s, w \leq 10\,000$). Они означают длины сторон, параллельных осям Ox и Oy соответственно. Во второй строке находится целое число N ($1 \leq N \leq 15\,000$), количество рудников. В последующих N строках находятся координаты рудников. В $i+2$ строке содержатся 2 целых числа x, y ($-30\,000 \leq x, y \leq 30\,000$), означающих координаты i -го рудника.

Формат выходных данных

В первой строке выходного файла нужно вывести максимальное количество рудников внутри площадки. Во второй строке нужно вывести координату угла любой из возможных оптимальных площадок. У прямоугольника четыре угла. Вам следует выводить тот, у которого x и y координаты максимальны. Выведенные координаты должны быть целыми и не должны превосходить 10^9 по абсолютной величине.

Пример

stdin	stdout
1 2	4
12	3 3
0 0	
1 1	
2 2	
3 3	
4 5	
5 5	
4 2	
1 4	
0 5	
5 0	
2 3	
3 2	

Задача F4. Внутренняя точка 2 [3.5 sec, 256 mb]

Дан совсем невыпуклый *простой* N -угольник и K точек. Напомним, N -угольник называется простым, если не имеет ни самопересечений, ни самокасаний. Для каждой точки нужно определить, где она находится — внутри, на границе, или снаружи.

Формат входных данных

В первой строке дано целое число T — количество тестов.

Далее идут T тестов. Тесты разделены переводом строки.

N ($3 \leq N \leq 10^5$). Далее N точек — вершины многоугольника.

K ($0 \leq K \leq 10^5$). Далее K точек — запросы.

Все координаты — целые числа по модулю не превосходящие 10^9 .

Суммарное количество N и K не превосходит $2 \cdot 10^6$.

Формат выходных данных

Для каждого запроса одна строка — INSIDE, BORDER или OUTSIDE.

Тесты следует разделять переводом строки.

Примеры

stdin	stdout
1	INSIDE
4	BORDER
0 0	BORDER
2 0	OUTSIDE
2 2	
0 2	
4	
1 1	
0 0	
0 1	
0 3	

Задача F5. Антиуфология [1.0 sec, 256 mb]

В далекой Англии члены общества нелюбителей уфологии решили устроить новую всемирную головоломку для своих антиколлег — уфологов. Суть ее — выжечь на поле параллельные линии так, чтобы оно оказалось разделено на равновеликие (одинаковые по площади) части. Поле представляет собой многоугольник без самопересечений, причем никакие его соседние (имеющие общую вершину) ребра не лежат на одной прямой.

Оборудование общества позволило смоделировать поле, и наложить на эту модель декартову сетку, тем самым, получив координаты всех вершин многоугольника. Из-за отсутствия финансового обеспечения со стороны правительства, мощность оборудования общества позволяет смоделировать искомые линии только горизонтальными (то есть параллельными оси абсцисс x).

Вам необходимо найти горизонтальные прямые, такие, чтобы они делили заданный многоугольник на равновеликие (возможно, несвязные) части.

Формат входных данных

Во входном файле в первой строке содержатся два числа: N и K , $3 \leq N \leq 50\,000$, $1 \leq K \leq 50\,000$, где N — это количество вершин многоугольника, а K — количество прямых, которые необходимо провести, чтобы разделить многоугольник на $(K + 1)$ равновеликие части.

Следующие N строк содержат вершины многоугольника, заданные в порядке обхода по часовой стрелке. Каждая вершина задается двумя координатами X и Y — числа в диапазоне от $-10\,000$ до $10\,000$.

Формат выходных данных

Выходной файл должен содержать K строк, в каждой из которых записана ордината соответствующей прямой с точностью до 0.0001 . Прямые в файле должны быть отсортированы в порядке возрастания ординат.

Пример

stdin	stdout
8 1 1 1 1 4 3 4 3 0 -3 0 -3 4 -1 4 -1 1	1.7500

Деревянные структуры

Задача G1. Отрезки, пересекающие прямоугольник [2 sec, 256 mb]

Будем считать, что отрезок и прямоугольник пересекаются, если они имеют хотя бы одну общую точку.

Вам дан набор из n горизонтальных и вертикальных отрезков на плоскости. Выполните q запросов. Каждый запрос представляет из себя прямоугольник, со сторонами параллельными осям координат, а ответом на запрос является множество отрезков, пересекающих этот прямоугольник.

Формат входных данных

В первой строке дано целое число n ($1 \leq n \leq 50\,000$) — количество отрезков.

В следующих n строках дано по четыре целых числа x_1, y_1, x_2, y_2 — координаты двух концов отрезка. Гарантируется, что все отрезки параллельны осям координат и ни один не вырожден в точку.

В следующей строке дано целое число q ($1 \leq q \leq 50\,000$) — количество запросов.

В следующих q строках даны по четыре целых числа x_1, y_1, x_2, y_2 ($x_1 \leq x_2, y_1 \leq y_2$) — координаты левого нижнего и правого верхнего угла прямоугольника из запроса.

Все координаты целые, по модулю не превосходят 10^6 .

Формат выходных данных

Пусть I_j — множество индексов отрезков, пересекающих j -й прямоугольник. Выведите медиану множества. Если $|I_j| \bmod 2 = 0$, выведите меньшее из двух чисел. Медиана пустого множества равна -1 . Отрезки нумеруются от 1 до n в порядке, в котором они идут во входных данных. Гарантируется, что $\sum_j |I_j| \leq 10^7$.

Примеры

stdin	stdout
3	2
0 0 0 5	2
-1 4 5 4	1
6 6 6 10	-1
4	
0 0 6 6	
2 2 4 5	
-1 -1 1 6	
-2 -2 -1 -1	

Замечание

В примере ответами на запросы являются следующие множества:

$\{1, 2, 3\}, \{2\}, \{1, 2\}, \{\}$.

Соответственно, ответы на запросы: 2, 2, 1, -1 .

Задача G2. Отрезки, пересекающие прямоугольник [6 сек, 256 mb]

Будем считать, что отрезок и прямоугольник пересекаются, если они имеют хотя бы одну общую точку.

Вам дан набор из n непересекающихся отрезков на плоскости. Выполните q запросов. Каждый запрос представляет из себя прямоугольник, со сторонами параллельными осям координат, а ответом на запрос является множество отрезков, пересекающих этот прямоугольник.

Формат входных данных

В первой строке дано целое число n ($1 \leq n \leq 50000$) — количество отрезков.

В следующих n строках дано по четыре целых числа x_1, y_1, x_2, y_2 — координаты двух концов отрезка. Все координаты по модулю не превосходят 10^6 . Гарантируется, что все отрезки не параллельны осям координат.

В следующей строке дано целое число q ($1 \leq q \leq 50000$) — количество запросов.

В следующих q строках даны по четыре целых числа x_1, y_1, x_2, y_2 ($x_1 \leq x_2, y_1 \leq y_2$) — координаты левого нижнего и правого верхнего угла прямоугольника из запроса.

Формат выходных данных

Для каждого запроса выведите число k — количество отрезков, пересекающих прямоугольник из запроса. Затем выведите номера этих отрезков без повторов в любом порядке. Отрезки нумеруются в порядке, в котором они идут во входных данных. Гарантируется, что суммарное число отрезков в ответе не превышает 10^6 .

Примеры

stdin	stdout
2	2 1 2
0 0 6 -1	1 2
2 2 4 5	1 2
3	
1 -1 3 2	
3 2 5 4	
1 1 5 6	

Герман и четырёхмерная модель теории относительности

Задача H1. Не лётная погода [0.3 sec, 256 mb]

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

stdin	stdout
4 4	540.0
400 400	
600 400	
600 600	
400 600	
0 0	
1000 0	
1000 1000	
0 1000	

Задача H2. Лоцман [0.5 sec, 256 mb]

Внимание, внимание, говорит Москва, работают все радиостанции... Передаём экстренное сообщение.

Сегодня состоятся последние ходовые испытания нового супертанкера, флагмана отечественного наливного флота «Нефтяник». Впервые в мировой практике будет продемонстрирована новая разработка наших учёных – система «Лоцман», позволяющая оптимально управлять кораблем без участия человека в акватории прямоугольной гавани. Корабль (а ведь все корабли имеют треугольную форму) умеет перемещаться по кратчайшему расстоянию из заданного начального положения в конечное между другими кораблями, стоящими на якоре в гавани. На «Нефтянике» было предусмотрено практически всё, что необходимо для безопасного испытания: и толстая стальная обшивка, и электронная система управления, и спутниковая связь с комплексом определения координат, и чуткий радар. Но в дело вмешался, как это обычно и бывает, человеческий фактор. Вовочка, сын командира корабля, тайком пробрался на секретный объект как раз перед презентацией, сел к бортовому компьютеру и решил скоротать время за любимой компьютерной игрой. В результате на компьютер проник вирус и испортил часть функций программы «Лоцман». Теперь корабль не может совершать повороты вокруг своей оси. Требуется составить программу, вычисляющую длину кратчайшего маршрута танкера к заданной точке.

Формат входных данных

В первой строке записаны целые числа DX, DY – размеры гавани ($0 < DX, DY < 10^8$). Строки 2-4 содержат по два целых числа – координаты танкера (напоминаем, что корабль «Нефтяник», как и все остальные корабли, имеет треугольную форму). В строке 5 указана точка, куда должен в итоге попасть корабль (а именно, тот из его углов, координаты которого описаны во второй строке входа). В строке 6 записано целое число N (количество других кораблей в гавани, $0 \leq N \leq 40$). Следующие $3N$ строк входа содержат координаты этих кораблей. Все координаты находятся в пределах гавани. Угол гавани является началом координат. Корабли не пересекаются.

Формат выходных данных

Выведите единственное число – минимальную длину маршрута, округлённую до трёх знаков после десятичной точки, либо значение -1 , если «Нефтянику» из-за повреждения навигационной программы не удастся доплыть до цели.

Примеры

stdin	stdout
2003 2003 20 50 10 30 30 30 140 60 1 80 1000 100 20 60 20	146.569

3D

Задача I. The Worm in the Apple [0.5 sec, 256 mb]

В 3D даны n точек и q точек-запросов. Постройте выпуклую оболочку n точек и для каждого из q запросов найдите расстояние до поверхности выпуклой оболочки.

Формат входных данных

На первой строке число n ($4 \leq n \leq 1,000$). Далее n строк содержат точки для выпуклой оболочки. Никакие четыре из них не лежат в одной плоскости. Далее число запросов q ($1 \leq q \leq 10^5$) и q строк, содержащих точки-запросы. Все точки запросы гарантированно внутри выпуклой оболочки. Все координаты целые, по модулю не более 10 000.

Формат выходных данных

Для каждого запроса выведите число с четырьмя знаками после запятой – расстояния до поверхности выпуклой оболочки.

Примеры

stdin	stdout
6	1.0000000000
0 0 0	2.8867513459
100 0 0	7.0000000000
0 100 0	2.0000000000
0 0 100	
20 20 20	
30 20 10	
4	
1 1 1	
30 30 35	
7 8 9	
90 2 2	