

Задача А. Пусти козла в огород - 1

Имя входного файла: `goat1.in`
Имя выходного файла: `goat1.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

64 мегабайта

Однажды на огород к Ивану Петровичу забежало целых три козла. Известно, что козлы весьма агрессивные животные, особенно когда речь идет о борьбе за вкусную капусту. Поэтому каждый из трех козлов, заметив других козлов, замер на месте и начал наблюдать за оставшимися козлами: одним глазом за одним козлом, другим — за другим. Естественно, для этого козлу нужно “косить” глазами.

Определите наибольший угол, на который пришлось “раскосить” глазами козлам.

Формат входных данных

Программа получает на вход координаты трех точки, в которых стоят козлы (сначала координаты первого козла, затем второго, затем третьего). Координаты — пара целых чисел, не превосходящих 10^4 по модулю.

Формат выходных данных

Выведите единственное число — величину самого большого угла, на который “косят” глаза козлов с точностью не меньше 6 знаков после запятой.

Пример

<code>goat1.in</code>	<code>goat1.out</code>
0 0 3 0 0 4	90.000000

Задача В. Пусти козла в огород - 5

Имя входного файла: `goat5.in`
Имя выходного файла: `goat5.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

64 мегабайта

В огород к Ивану Петровичу повадился чужой козел по ночам воровать капусту. Чтобы поймать наглеца, Иван Петрович установил на огороде прожектор, освещающий часть плоскости в форме некоторого угла. И когда очередной ночью Иван Петрович слышал хрумканье в своем огороде, он включил свой прожектор. Определите, увидит ли Иван Петрович чужого козла или нет.

Формат входных данных

Программа получает на вход координаты четырех точек A , O , B , P . Прожектор установлен в точке O , точки A и B лежат на границах освещенной прожектором области (на разных лучах), в точке P находится козел. Все числа во входном файле целые и не превышают 100 по абсолютному значению. Точки A , O и B не лежат на одной прямой.

Формат выходных данных

Выведите слово `YES`, если Иван Петрович увидит козла или слово `NO` в противном случае

Примеры

<code>goat5.in</code>	<code>goat5.out</code>
0 1 0 0 1 0 1 1	YES
1 0 0 0 0 1 -1 -1	NO

Задача D. Биссектриса

Имя входного файла: `bisector.in`
Имя выходного файла: `bisector.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

64 мегабайта

Найдите прямую, содержащую биссектрису угла, заданного вершиной X и двумя точками Y и Z на его сторонах.

Формат входных данных

Шесть целых чисел, не превышающих по модулю 10^4 — координаты точек X , Y и Z .

Формат выходных данных

Три числа — коэффициенты нормального уравнения биссектрисы угла $\angle YXZ$ с точностью до шести знаков после запятой.

Пример

<code>bisector.in</code>	<code>bisector.out</code>
1 1 1 0 0 1	-1.0 1.0 -0.0

Задача Е. Пусти козла в огород - 4

Имя входного файла: `goat4.in`
Имя выходного файла: `goat4.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

64 мегабайта

Огород у Петра Васильевича имеет форму треугольника и при этом не огорожен. Петр Васильевич хочет привязать козла к колышку в огороде так, чтобы он мог пастись на участке огорода максимальной площади, но при этом не заходил бы на участки соседей. Определите, в каком месте огорода нужно привязать козла.

Формат входных данных

Программа получает на вход координаты вершин трёх углов огорода. Координаты — пара целых чисел, не превосходящих 10^4 по модулю.

Формат выходных данных

Выведите два числа — координаты колышка, к которому нужно привязать козла, с точностью не менее 6 знаков после запятой.

Пример

<code>goat4.in</code>	<code>goat4.out</code>
0 0 3 0 0 4	1.00000000 1.00000000

Задача F. Пусти козла в огород - 3

Имя входного файла: `goat3.in`
Имя выходного файла: `goat3.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

64 мегабайта

Козла пустили в огород. К несчастью для козла, его крепко-накрепко привязали к точке C крепчайшей альпинистской верёвкой длины L (то есть козёл может кушать травку, удалённую от точки C не более чем на L метров). И только одно скрасило жизнь бедного козла — недалеко от него виднеется грядка так любимых им ананасов, представляющая собой отрезок прямой с концами в точках A и B . Но, при большом старании, козёл может растянуть крепчайшую альпийскую верёвку. Определите, на какую длину козлу придётся растянуть верёвку, чтобы добраться хотя бы до одного ананасика? А чтобы съесть все ананасы?

Формат входных данных

На входе находятся координаты точек A , B , C и длина верёвки L . Все числа целые, $L \geq 0$, все координаты не превосходят по модулю 10000. Числа разделены пробелами или переводами строк.

Формат выходных данных

В первой строке вывести минимальную длину, на которую козлу придётся растянуть верёвку, чтобы дотянуться до грядки с ананасами. Во второй строке вывести минимальную длину, на которую козлу придётся растянуть верёвку, чтобы добраться до всех ананасов с грядки. Все числа выводить с точностью не менее 6 знаков после запятой.

Пример

<code>goat3.in</code>	<code>goat3.out</code>
8 -6 8 6	1.0000000000
0 0 7	3.0000000000

Задача G. Пусти козла в огород - 2

Имя входного файла: `goat2.in`
Имя выходного файла: `goat2.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

64 мегабайта

Козла пустили в квадратный огород и привязали к колышку. Колышек воткнули точно в центре огорода. Козёл голоден, как волк, прожорлив, как бык, и ест всё, до чего дотянется, не перелезая через забор и не разрывая веревку. Какая площадь огорода будет объедена?

Формат входных данных

Длина стороны огорода и длина верёвки в метрах (положительные целые числа, не превосходящие 100, расположенные в одной строке через пробел).

Формат выходных данных

Площадь части огорода (в квадратных метрах, с точностью до 6 знаков после десятичной точки), объеденной козлом.

Пример

<code>goat2.in</code>	<code>goat2.out</code>
10 6	95.091113

Задача Н. Пусти козла в огород - 6

Имя входного файла: `goat6.in`
Имя выходного файла: `goat6.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

64 мегабайта

Петр Васильевич в ярости! Ведь сосед Василий Петрович выгуливал козла в его огороде! Как не предусмотрителен был Василий Петрович — ведь у Петра Васильевича целых 2 козла и оба они в ответ будут поедать и вытаптывать соседский огород. Огород Василия Петровича большой и неогороженный, в некоторых его местах растут деревья. Козлам потребуется много времени, чтобы выполнить свою миссию. Поэтому Петр Васильевич решил привязать каждого козла к какому-нибудь дереву, и пусть себе гуляют. Но привязать каждого надо так чтобы он не доставал до всех деревьев кроме того, к которому он привязан, иначе он запутается в веревке. Кроме того, надо чтобы они не доставали друг до друга, иначе они будут вытаптывать одну и ту же территорию. Чтобы нанести максимальный вред своему соседу, Петр Васильевич хочет, чтобы суммарная площадь, доступная козлам была максимальна. Но нельзя привязывать козла на расстоянии меньше 1 метра от дерева и дальше, чем на 50 метров.

Формат входных данных

В первой строке записано целое число $N (2 \leq N \leq 1000)$ — количество деревьев в огороде. В следующих N строках записаны координаты деревьев. Начало координат совмещено с центром огорода, координаты даны в метрах с точностью до сантиметра. Координаты деревьев по модулю не превосходят 100 метров. Можно считать, что нельзя привязать козла так, чтобы он смог выйти за пределы огорода. Размерами самих козлов можно пренебречь. Гарантируется, что козлов всегда можно привязать надлежащим образом.

Формат выходных данных

Выведите максимальную площадь, которую смогут вытоптать козлы Петра Васильевича, с точностью не менее 6 знаков после запятой.

Пример

<code>goat6.in</code>	<code>goat6.out</code>
8 1 1 -2 0 5 3 -2 3 8 3.10 -2 -1 -2 2 8 4.10	36.8060473804

Задача I. Пусти козла в огород -7

Имя входного файла: `goat7.in`
Имя выходного файла: `goat7.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

64 мегабайта

Петр Васильевич всегда выгуливает своих козлов у себя в огороде, но ему не нравится, что они вытаптывают его грядки, поэтому он решил усовершенствовать привязь. Теперь ошейник у них прикреплен к веревке, натянутой между двумя деревьями, поэтому козлы могут ходить только по отрезку, их соединяющему.

Но козлам скучно ходить по огороду по одиночке, поэтому они хотят встретиться и пожевать траву вместе. Помогите им встретиться!

Формат входных данных

Программа получает на вход восемь целых чисел, по абсолютной величине не превосходящих 10^4 . Сначала указываются координаты двух деревьев, к которым привязан первый козел, а затем - координаты деревьев, к которым привязан второй. Заметим, что для привязи могут служить одни и те же деревья (в том числе могут совпадать и концы одной привязи).

Формат выходных данных

Если козлам встретиться не суждено, выведите строку `Empty`. Если они могут встретиться только в одной точке, то выведите два числа — координаты точки пересечения. Если пересечением является отрезок, то выведите четыре числа — координаты двух концов отрезка в лексикографическом порядке (то есть сначала нужно вывести ту точку, у которой меньше координата x , а если у них равны координаты x то ту, у которой меньше координата y). Все числа следует выводить с точностью не менее 6 знаков после запятой.

Примеры

<code>goat7.in</code>	<code>goat7.out</code>
0 0 9 9 9 5 0 5	5.0000000000 5.0000000000
0 0 9 9 15 15 7 7	7.0000000000 7.0000000000 9.0000000000 9.0000000000
0 0 9 9 10 10 10 10	Empty

Задача J. Месть дракона

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Один сказочный король до смерти ненавидел драконов. Он собрал огромную армию и убил их всех до единого. Но он упустил из вида одно небольшое драконье яйцо, приняв его за камень, так что из этого яйца вылупился последний дракон. Быстро установив виновника геноцида остальных драконов, последний дракон сильно разозлился и разрушил всё королевство. Армии, чтобы от него отбиваться, совсем не осталось, поскольку она вся была потрачена на уничтожение всех остальных драконов.

Сказочное королевство представляет собой координатную плоскость. Король с остатками армии укрылся в каменном дворце, который находится в точке с координатами (x_p, y_p) . Дворец надёжно защищён, и дракон не способен его разрушить. Поняв это, дракон спрятался в точке с координатами (x_d, y_d) в надежде подкараулить короля, когда тот покинет дворец. Король вскоре узнал о местонахождении дракона и теперь пытается оценить опасность ситуации.

Если король будет находиться за пределами дворца, то в любой момент дракон может это заметить и вылететь за ним из своего логова. Вылет дракона немедленно будет замечен стражей, и король тут же начнёт равномерно прямолинейно двигаться в сторону дворца, пока не окажется в безопасности за его стенами. Дракон же жаждет мести, поэтому просто в каждый момент времени будет равномерно прямолинейно двигаться точно в направлении текущего местоположения короля. Король уже стар, а дракон в самом расцвете сил, так что скорость перемещения короля ровно в два раза меньше, чем скорость перемещения дракона. Король хотел бы определить, насколько сильно дракон ущемляет свободу его передвижения, поэтому желает узнать площадь безопасного участка плоскости, по которому он может разгуливать без опасений, что в случае вылета дракон догонит его раньше, чем он доберётся до дворца.

Формат входных данных

Первая строка содержит два целых числа через пробел: x_p и y_p ($-1000 \leq x_p, y_p \leq 1000$) — координаты дворца.

Вторая строка содержит два целых числа через пробел: x_d и y_d ($-1000 \leq x_d, y_d \leq 1000$) — координаты логова дракона.

Эти две точки не совпадают.

Формат выходных данных

Выведите единственное вещественное число — площадь безопасной части плоскости. Абсолютная или относительная погрешность не должна превышать 10^{-6} .

Пример

стандартный ввод	стандартный вывод
0 0	0.916297857297023
1 0	

Задача К. Скользящая симметрия

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

Движением плоскости называют такое преобразование плоскости, которое сохраняет попарные расстояния между точками, то есть если A_1 и B_1 – образы некоторых точек A и B при движении, то $|A_1B_1| = |AB|$.

Одной из разновидностей движения плоскости является скользящая симметрия. Скользящей симметрией называют композицию симметрии относительно некоторой прямой l и переноса на вектор, параллельный l (этот вектор может быть нулевым). На рисунке (нет рисунка) показан пример применения скользящей симметрии к отрезку.

Известно, что любой отрезок можно перевести в любой другой отрезок такой же длины с помощью скользящей симметрии. Требуется по координатам двух различных точек A и B и двух точек A_1 и B_1 , находящихся на таком же расстоянии друг от друга, как и точки A и B , найти скользящую симметрию, переводящую точку A в точку A_1 , а точку B в точку B_1 .

Формат входных данных

В первой строке входного файла находятся четыре целых числа – координаты двух различных точек A и B . Во второй строке также находятся четыре целых числа – координаты двух различных точек A_1 и B_1 . Гарантируется, что $|A_1B_1| = |AB|$. Все числа во входном файле по модулю не превышают 1000. Числа в строках разделены пробелом.

Формат выходных данных

Выведите в выходной файл описание искомой скользящей симметрии, которое представляется в следующем виде.

В первой строке должны выводиться координаты двух различных точек, лежащих на прямой l , относительно которой выполняется симметрия, а во второй – координаты вектора, параллельного этой прямой, на который осуществляется перенос. Вещественные числа должны быть представлены не менее чем с 6 знаками после десятичной точки.

Примеры

стандартный ввод	стандартный вывод
1 1 3 2 -1 1 -3 2	0.0000000 1.0000000 0.0000000 3.0000000 0.0000000 0.0000000
1 1 3 1 3 -1 5 -1	2.0000000 0.0000000 6.0000000 0.0000000 2.0000000 0.0000000

Задача L. Точка в многоугольнике

Имя входного файла: `point.in`
Имя выходного файла: `point.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

64 мегабайта

Формат входных данных

В первой строке содержится три числа — N ($3 \leq N \leq 100000$) и координаты точки. Последующие N строк содержат координаты углов многоугольника. Координаты — целые, не превосходят 10^6 по модулю.

Формат выходных данных

Одна строка YES, если заданная точка содержится в приведённом многоугольнике или на его границе, и NO в противном случае.

Пример

<code>point.in</code>	<code>point.out</code>
3 2 3 1 1 10 2 2 8	YES

Задача М. Выпукл ли многоугольник

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

Пришли гномы и утащили забавную легенду к этой задаче.

Формат входных данных

В первой строке вводится одно число N ($3 \leq N \leq 100\,000$). Далее в N строках задается по паре чисел – координаты очередной вершины простого многоугольника в порядке обхода по или против часовой стрелки.

Формат выходных данных

Выведите одну строку: «YES», если приведённый многоугольник является выпуклым, и «NO» в противном случае.

Примеры

стандартный ввод	стандартный вывод
3 0 0 0 1 1 0	YES
6 0 0 0 2 1 2 1 1 2 1 2 0	NO

Задача N. Площадь многоугольника

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

На плоскости задан многоугольник координатами вершин в порядке их обхода. Многоугольник не обязательно выпуклый. Требуется найти его площадь.

Формат входных данных

Сначала вводится число N – количество вершин многоугольника ($3 \leq N \leq 100\,000$), затем N пар целых чисел, задающих координаты его вершин. Все координаты по модулю не превосходят 10^9 .

Формат выходных данных

Выведите площадь многоугольника с минимальным необходимым (вероятно, нулевым) количеством знаков после запятой.

Пример

стандартный ввод	стандартный вывод
3 1 0 1 4 7 4	12

Задача О. Выпуклая оболочка

Имя входного файла:	<code>hull.in</code>
Имя выходного файла:	<code>hull.out</code>
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Дано N точек на плоскости.

Нужно построить их выпуклую оболочку.

Гарантируется, что выпуклая оболочка не вырождена.

Формат входных данных

На 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. Кроме того, в последней строке выведите площадь получившейся выпуклой оболочки. Площадь необходимо вывести абсолютно точно.