

Задача А. Расстояние от точки до отрезка

Имя входного файла: distance4.in
Имя выходного файла: distance4.out
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Найдите расстояние от заданной точки до заданного отрезка.

Формат входных данных

Шесть целых чисел — координаты точки и координаты концов отрезка.

Формат выходных данных

Одно число — расстояние от точки до отрезка с точностью не менее 10^{-6} .

Пример

distance4.in	distance4.out
0 4 2 3 2 5	2.0

Задача В. Пересечение двух прямых

Имя входного файла: `intersec1.in`
Имя выходного файла: `intersec1.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Формат входных данных

Шесть чисел — коэффициенты A , B и C нормального уравнения двух различных непараллельных прямых (сначала для одной прямой, затем для другой).

Формат выходных данных

Два числа — координаты точки их пересечения.

Пример

<code>intersec1.in</code>	<code>intersec1.out</code>
1 1 -1 1 -1 0	0.5 0.5

Задача С. Пересечение двух отрезков

Имя входного файла: `intersec2.in`
Имя выходного файла: `intersec2.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Формат входных данных

Восемь чисел — координаты концов двух отрезков.

Формат выходных данных

Одна строка «YES», если отрезки имеют общие точки, и «NO» в противном случае.

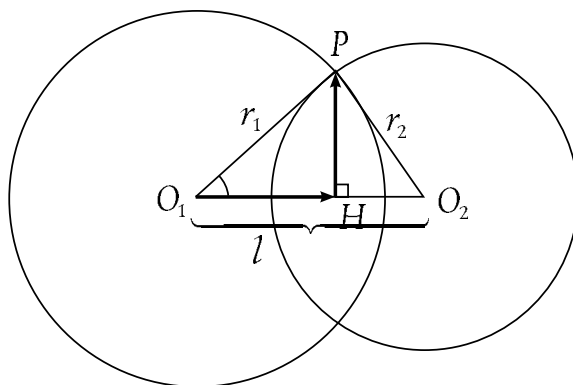
Пример

<code>intersec2.in</code>	<code>intersec2.out</code>
5 1 2 6 1 1 7 8	YES

Задача D. Две окружности

Имя входного файла:	intersec4.in
Имя выходного файла:	intersec4.out
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Вам заданы две окружности. Необходимо выяснить, пересекаются ли заданные окружности и найти точки их пересечения.



Формат входных данных

Первая строка файлов содержит количество тестов — целое число от 1 до 10 000.

Далее следуют тесты. Каждый тест имеет вид:

Две строки, по три числа в каждой — координаты центра и радиус сначала первой, а затем второй окружности.

Все числа — целые, по модулю не превосходящие 10 000. А радиус еще и положительный.

Формат выходных данных

На каждый тест выведите ответ в следующем формате:

На отдельной строке выходного файла выведите количество точек пересечения (0, 1, 2 или 3, если их бесконечно много). В случае одной точки пересечения выведите во второй строке координаты этой точки. В случае двух точек пересечения выведите во второй строке координаты точки H , в третьей длины векторов \overrightarrow{OH} и \overrightarrow{HP} , в следующих двух строках должны находиться координаты точек пересечения. Эти две точки можно вывести в произвольном порядке.

Замечание

Ответ однозначен. Координаты и длины нужно вывести с не менее чем 6-ю верными знаками после запятой. Заметим, что все входные данные — целые числа, поэтому величина “сколько точек пересечения” может быть определена точно.

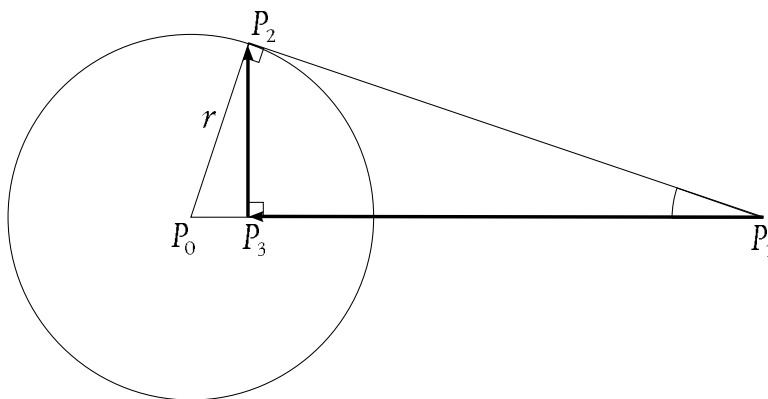
Пример

intersec4.in	intersec4.out
4	0
3 4 5	1
11 4 2	8.0000000000 4.0000000000
3 4 5	2
11 4 3	7.5625000000 4.0000000000
3 4 5	4.5625000000 2.0453835215
11 4 4	7.5625000000 6.0453835215
3 4 5	7.5625000000 1.9546164785
3 4 5	3

Задача Е. Касательная к окружности

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

Вам задана окружность и точка. Точка может лежать вне окружности, на ее границе, а также внутри окружности. Необходимо провести касательные к окружности (если это возможно) и найти точки касания.



Формат входных данных

В первой строке входного файла находятся два числа — координаты центра окружности P_0 . Во второй строке записан радиус окружности r . В третьей строке находятся два числа — координаты точки P_1 .

Формат выходных данных

В первой строке выходного файла выведите количество точек касания (0, 1 или 2). В случае одной точки касания выведите во второй строке координаты этой точки. В случае двух точек касания выведите во второй строке координаты точки P_3 , в третьей длины векторов $\overrightarrow{P_1P_3}$ и $\overrightarrow{P_3P_2}$, в следующих двух строках должны находиться координаты точек касания.

Пример

tangent.in	tangent.out
2 2	2
2	2.000000 3.333333
2 5	1.666667 1.490712
	3.490712 3.333333
	0.509288 3.333333

Задача F. Площадь многоугольника

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

Формат входных данных

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

Все координаты — целые числа, по модулю не превосходящие 10^4 .

Формат выходных данных

Одно число — величина площади приведённого многоугольника.

Пример

<code>area.in</code>	<code>area.out</code>
3 1 0 0 1 1 1	0.5

Задача G. Выпуклый многоугольник

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

Формат входных данных

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

Формат выходных данных

Одна строка «YES», если приведённый многоугольник является выпуклым, и «NO» в противном случае.

Пример

<code>polygon.in</code>	<code>polygon.out</code>
3 1 0 0 1 1 1	YES

Задача Н. Выпуклая оболочка

Имя входного файла: `hull.in`
Имя выходного файла: `hull.out`
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

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

<code>hull.in</code>	<code>hull.out</code>
5	4
0 0	0 0
2 0	2 0
0 2	2 2
1 1	0 2
2 2	

Задача I. Gifts Large and Small

Имя входного файла:	gift.in
Имя выходного файла:	gift.out
Ограничение по времени:	0.1 second
Ограничение по памяти:	256 megabytes

WrapIt.com specializes in wrapping gifts. Started several years ago as a service offered to local department stores and malls, today WrapIt serves customers world-wide and boasts that it can package anything from half-carat diamonds to whole apartment blocks. WrapIt has found that some customers prefer their gifts to be wrapped in the smallest possible packages, whereas others prefer large packages that make their gifts seem larger than they really are. The company needs a program that computes the smallest and largest rectangular package into which a gift can be "tightly" wrapped. Since this is a difficult problem, the company will initially settle for a two-dimensional version of the program. Each gift is approximated as a simple polygon, and all packages are represented by rectangles. A gift is said to "fit tightly" in a package if the gift touches all four sides of the package. For each gift, your program must compute the areas of the smallest and largest packages into which the gift can fit tightly.

Формат входных данных

The input contains several gift descriptions. Each description begins with a line containing an integer n ($3 \leq n \leq 100$), which is the number of vertices in the polygon that represents the gift. The following n lines contain pairs of integers that represent the coordinates of the polygon vertices, in clockwise order. Each polygon will have a non-zero area and will not intersect itself. The input is terminated by a line containing the integer 0.

Формат выходных данных

For each gift, first print the number of the gift. Then on separate lines, print the minimum and maximum areas of the packages into which the gift fits tightly, using the format in the sample output. Print a blank line after each test case. The computed areas should be exact to three digits to the right of the decimal point.

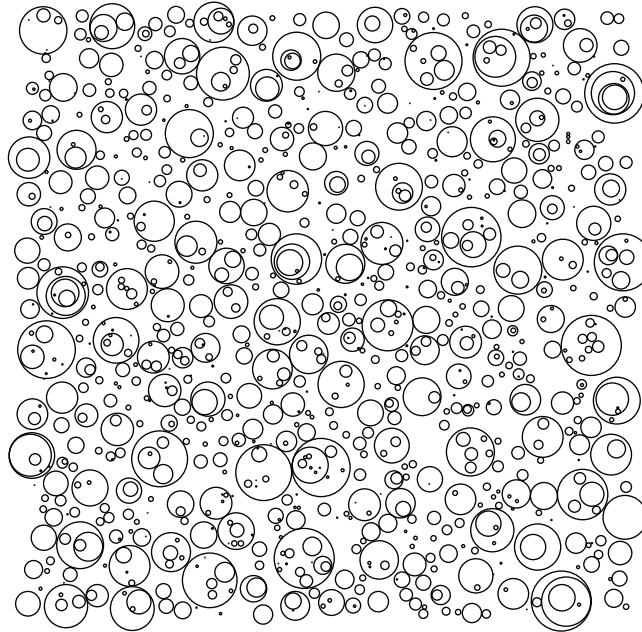
Пример

gift.in	gift.out
3	Gift 1
-3 5	Minimum area = 80.000
7 9	Maximum area = 200.000
17 5	
4	Gift 2
10 10	Minimum area = 100.000
10 20	Maximum area = 200.000
20 20	
20 10	
0	

Задача J. Circles

Имя входного файла: `circles.in`
 Имя выходного файла: `circles.out`
 Ограничение по времени: 5 seconds
 Ограничение по памяти: 256 megabytes

There are n circles located on the plane. Circles may have common points, but for any two circles their intersection is either a point, or one of the two circles.



Find the total area covered by at least one circle.

Формат входных данных

The first line of the input file contains integer number n ($1 \leq n \leq 100\,000$). The following n lines contain three integers each and describe circles. The i -th circle is described by coordinates of its center x_i and y_i and its radius r_i ($-10^6 \leq x_i, y_i \leq 10^6$, $1 \leq r_i \leq 10^6$).

Формат выходных данных

Output one real number: the total area covered by at least one circle. Your answer must have absolute or relative error of at most 10^{-9} .

Пример

<code>circles.in</code>	<code>circles.out</code>
4 2 2 2 2 2 1 5 2 1 5 5 2	28.2743338823081391

Задача К. Выходной

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

Как известно, следующий день после олимпиады является выходным днем. Однако толпы ЛКШат стремятся попасть в комповник даже в выходной день. Преподаватели решили запретить ЛКШатам приближаться к комповнику. Чтобы ограничить доступ к комповнику жителям определенного домика, проводится линия, разделяющая домик и комповник, которую запрещено пересекать жителям этого домика. Разумеется домик лежит по одну сторону линии, а комповник — по другую.

Будем считать, что как домик, так и комповник представляют собой выпуклые многоугольники на плоскости. Линия, их разделяющая — не что иное, как прямая на плоскости. Многоугольники, соответствующие домику и комповнику, лежат по разные стороны от этой прямой.

Ваша задача — по данным координатам вершин этих многоугольников найти прямую их разделяющую.

Формат входных данных

В первой строке входного файла задано число N ($1 \leq N \leq 50\,000$) — количество вершин в многоугольнике, соответствующем домику. Затем в N строках перечислены координаты вершин многоугольника по часовой стрелке. Затем на отдельной строке задано число M ($1 \leq M \leq 50\,000$) — количество вершин в многоугольнике, соответствующем комповнику. Затем в M строках перечислены координаты вершин этого многоугольника по часовой стрелке. Все координаты целые и по модулю не превышают 10^9 . В обоих многоугольниках никакие три точки не лежат на одной прямой.

Формат выходных данных

Выведите в файл три числа A, B, C — коэффициенты прямой, описываемой уравнением $Ax + By + C = 0$. Прямая должна быть нормированной, то есть должно быть выполнено следующее: $\sqrt{A^2 + B^2} = 1$. Описанные во входных данных многоугольники должны лежать по разные стороны от выведенной прямой и не должны пересекаться с ней.

Пример

rest.in	rest.out
4	1 0 -1.5
0 0	
0 1	
1 1	
1 0	
3	
2 0	
2 2	
4 0	

Задача L. Перестановки

Имя входного файла: `permutation.in`
Имя выходного файла: `permutation.out`
Ограничение по времени: 0.5 секунд
Ограничение по памяти: 256 мегабайт

Вася выписал на доске в каком-то порядке все числа от 1 по N , каждое число ровно по одному разу. Количество чисел оказалось довольно большим, поэтому Вася не может окинуть взглядом все числа. Однако ему надо всё-таки представлять эту последовательность, поэтому он написал программу, которая отвечает на вопрос — сколько среди чисел, стоящих на позициях с x по y , по величине лежат в интервале от k до l . Сделайте то же самое.

Формат входных данных

В первой строке лежит два натуральных числа — $1 \leq N \leq 100\,000$ — количество чисел, которые выписал Вася и $1 \leq M \leq 100\,000$ — количество вопросов, которые Вася хочет задать программе. Во второй строке дано N чисел — последовательность чисел, выписанных Васей. Далее в M строках находятся описания вопросов. Каждая строка содержит четыре целых числа $1 \leq x \leq y \leq N$ и $1 \leq k \leq l \leq N$.

Формат выходных данных

Выведите M строк, каждая должна содержать единственное число — ответ на Васин вопрос.

Пример

permutation.in	permutation.out
4 2	1
1 2 3 4	3
1 2 2 3	
1 3 1 3	

Задача М. Прямоугольные запросы

Имя входного файла: `rectmin.in`
 Имя выходного файла: `rectmin.out`
 Ограничение по времени: 5 секунд
 Ограничение по памяти: 256 мегабайт

Даны N точек на плоскости, у каждой точки есть ценность. Нужно быстро обрабатывать запросы двух типов:

- Присвоить всем точкам в области $[x_1..x_2] \times [y_1..y_2]$ ценность K .
- Найти точку с минимальной ценностью в области $[x_1..x_2] \times [y_1..y_2]$.

Формат входных данных

Число точек N ($1 \leq N \leq 262144$) и N точек. Каждая точка задается тремя числами — x , y , начальная ценность.

Число запросов M ($1 \leq M \leq 10^4$) и M запросов в формате `= x_1 y_1 x_2 y_2 $value$` для присваивания и `? x_1 y_1 x_2 y_2` для взятия минимума.

Все координаты от -10^9 до 10^9 . Все ценности от 0 до 10^9 .

Формат выходных данных

На каждый запрос `?` выведите минимальную ценность точек в прямоугольнике. Если в прямоугольнике нет ни одной точки, выведите `NO`.

Пример

rectmin.in	rectmin.out
4	2
1 1 1	1
-1 1 1	NO
-1 -1 1	0
1 -1 1	
7	
= 0 0 3 3 2	
= -3 -3 0 0 2	
? 0 0 3 3	
? -3 -3 3 3	
= -1 -1 1 1 0	
? 0 0 0 0	
? -1000 -1000 1000 1000	

Задача N. Кто ходит в гости по утрам

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

В Чудесном Лесу живут N различных персонажей, у каждого из которых есть свой собственный домик. Следуя заветам одного из самых известных лесных персонажей, Винни-Пуха, каждый житель считает необходимым проснуться с утра пораньше, умыться, одеться и пойти в гости к кому-либо. Разумеется, чтобы поступить не просто мудро, а очень мудро и не потратить слишком много времени на дорогу, персонаж отправится не к кому-нибудь, а к своему соседу, то есть к тому из жителей, домик которого находится к данному персонажу на наименьшем возможном расстоянии. Нетрудно понять, что хозяина этого домика не окажется дома, поскольку он тоже воспользуется правилом Винни-Пуха. Лишь по этой причине некому будет ни крикнуть “Ура!”, ни обрадоваться гостям. Если вдруг окажется, что несколько домиков расположены на минимальном расстоянии от персонажа, то он выберет для похода в гости домик с наименьшим номером. Ваша задача – определить какие персонажи соберутся у каждого домика.

Формат входных данных

В первой строке задается количество персонажей N ($2 \leq N \leq 100\,000$). В каждой из последующих N строк задаются по два числа – координаты точки на плоскости, в которой расположен домик соответствующего персонажа. Все координаты – целые неотрицательные числа, не превосходящие 10^9 .

Формат выходных данных

Выведите N строк. i -ая строка должна содержать число i , за которым следует двоеточие и далее в порядке возрастания номера персонажей, которые придут в гости в i -ый домик.

Пример

k.in	k.out
6	1: 2 3
0 0	2: 1
1 0	3:
0 1	4: 5
3 3	5: 4 6
2 2	6:
3 1	