

Задача А. Длина вектора

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

Вычислить длину вектора $\vec{a} = (x, y, z)$

Формат входных данных

Первая строка входного файла содержит три вещественных числа – координаты x, y, z , разделённые пробелами.

Формат выходных данных

Первая строка выходного файла должна содержать длину вектора с точностью до 2 знаков после десятичной точки.

Примеры

<code>vector.in</code>	<code>vector.out</code>
1 2 2	3.00
4.6 2.7 1.1	5.45

Задача В. Прямая по 2-м точкам

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

Даны координаты 2 точек на плоскости, выведите коэффициенты (a, b, c) прямой проходящей через эти точки. Прямая задана уравнением: $ax + by + c = 0$.

Формат входных данных

В первой строке входного файла $x_1 y_1 x_2 y_2$ — координаты точек разделенные пробелами (точки могут совпадать). Координаты — вещественные числа с точностью до 10^{-3} .

Формат выходных данных

Вывести $a b c$ - коэффициенты прямой (с точностью до 10^{-3}). $a = b = c = 0$ только если точки совпадают. При проверке координаты точек подставляются в выведенную вами прямую. Ответ считается корректным, если уравнение выполняется с точностью до 10^{-3} .

Примеры

<code>line.in</code>	<code>line.out</code>
1.0 0.0 2.0 0.0	0.000 1.000 -0.000

Задача С. Окружность

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

Дан радиус окружности, вычислить диаметр, длину и площадь.

Формат входных данных

В первой строке входного файла R - радиус окружности $10^{-3} \leq R \leq 10^6$) максимум с 3-знаками после десятичной точки.

Формат выходных данных

Вывести диаметр, длину и площадь окружности с точность до 3-х знаков.

Примеры

<code>radius.in</code>	<code>radius.out</code>
20	40.000 125.664 1256.637
3.234	6.468 20.320 32.857
7.999	15.998 50.259 201.012

Задача D. Сложение векторов

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

Дано 2 вектора в N -мерной системе счисления, нужно вывести их сумму.

Формат входных данных

В первой строке входного файла N - размерность векторов $1 \leq N \leq 100$).

Во второй строке N действительных чисел - компоненты вектора a , $(-10^5 \leq a_i \leq 10^5)$ максимум с 3-знаками после десятичной точки.

В третьей строке тоже N действительных чисел - компоненты вектора b , $(-10^5 \leq b_i \leq 10^5)$ максимум с 3-знаками после десятичной точки.

Формат выходных данных

Вывести N действительных чисел - компоненты суммы (с точностью до 3 знаков после десятичной точки).

Примеры

sum.in	sum.out
2 1 4 4 -5	5.000 -1.000
2 1 7 4 -5	5.000 2.000
3 3.992 7.451 8.326 6.014 2.144 3.772	10.006 9.595 12.098

Задача Е. Скалярное произведение

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

Дано 2 вектора в N -мерной системе координат, нужно вывести их скалярное произведение.

Формат входных данных

В первой строке входного файла N - размерность векторов ($1 \leq N \leq 100$).

Во второй строке N действительных чисел - компоненты вектора a , ($-10^7 \leq a_i \leq 10^7$) максимум с 3-знаками после десятичной точки.

В третьей строке тоже N действительных чисел - компоненты вектора b , ($-10^7 \leq b_i \leq 10^7$) максимум с 3-знаками после десятичной точки.

Формат выходных данных

Вывести одно действительное число - скалярное произведение (с точностью до 10^{-3}).

Примеры

product.in	product.out
2 1 4 4 -5	-16.000
2 1 7 4 -5	-31.000
3 3.99 7.4 8.32 6.01 2.1 3.77	70.886

Задача F. Треугольник

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

Дан прямоугольник $p \times q$ метров. Найдите треугольник максимально возможной площади, содержащийся внутри него.

Формат входных данных

В первой строке входного файла заданы два целых числа p и q через пробел — длины сторон прямоугольника ($1 \leq p, q \leq 100$).

Формат выходных данных

В первой строке выходного файла выведите одно число — максимальную площадь треугольника. В следующих трёх строках выведите по два числа x_i и y_i в каждой через пробел — координаты вершин треугольника, имеющего максимальную площадь. Порядок вывода вершин значения не имеет. Координаты должны удовлетворять соотношениям $0 \leq x_i \leq p$, $0 \leq y_i \leq q$. Все числа следует вывести не менее чем с шестью точными знаками после запятой. Если треугольников максимально возможной площади несколько, можно выводить любой из них.

Пример

triangle.in	triangle.out
1 1	0.5 0.633333 1.000000 0.000000 0.000000 1.000000 0.000000

Задача G. Две окружности

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

Даны координаты центров и радиусы окружностей, определить как они взаимно расположены на плоскости.

Формат входных данных

В первой строке входного файла $x_1 y_1 x_2 y_2 R_1 R_2$ - разделенные пробелами.
 $x_1 y_1 R_1$ - координаты центра и радиус первой окружности.
 $x_2 y_2 R_2$ - координаты центра и радиус второй окружности.

Формат выходных данных

Вывести в выходной файл одну из следующих строк: **Tangent: outside** - если касаются и при этом ни одна не лежит внутри другой, **Tangent: 1 in 2** - если касаются и первая лежит внутри второй, **Tangent: 2 in 1** - если касаются и вторая лежит внутри первой, **1 inside 2** - если первая лежит целиком внутри второй (не касаются), **2 inside 1** - если вторая лежит целиком внутри первой (не касаются), **Too far** - если окружности не касаются и не лежат внутри друг друга, **Intersect** - если окружности пересекаются в 2-х точках.

Примеры

twoc.in	twoc.out
1 1 4 5 3 2	Tangent: outside
1 1 2 1 3 2	Tangent: 2 in 1
2 1 1 1 2 3	Tangent: 1 in 2
1 1 2 1 3 1	2 inside 1
2 1 1 1 1 3	1 inside 2
2 1 9 1 1 3	Too far
2 1 1 1 3 3	Intersect

Задача H. Треугольники

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

На плоскости даны N точек. Никакие две точки не совпадают, никакие три не лежат на одной прямой. Найдите треугольник с вершинами в этих точках,

имеющий наименьший возможный периметр.

Формат входных данных

Во входном файле записано сначала число N - количество точек ($3 \leq N \leq 50$), а затем N пар вещественных чисел, задающих координаты точек.

Формат выходных данных

В выходной файл выведите три числа - номера точек, которые должны быть вершинами треугольника, чтобы его периметр был минимален. Если решений несколько выведите любое из них.

Примеры

triangles.in	triangles.out
5 0 0 1.3 0 -2 0.1 1 0 10 10	1 2 4

Задача I. Наибольшее расстояние

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

В некоторой стране N городов и M дорог. Каждая дорога представляет собой отрезок прямой, соединяющий два различных города, причём известно, что ни на одной дороге нет других городов, кроме тех, что задают её концы. Расстоянием между двумя городами считается минимальная суммарная длина дорог, по которым надо пройти, чтобы попасть из одного города в другой. Требуется вычислить наибольшее из таких расстояний.

Формат входных данных

В первой строке входного файла задано число N ($2 \leq N \leq 100$). В последующих N строках содержатся пары целых чисел $X_i Y_i$ ($-1000 \leq X_i, Y_i \leq 1000$), задающие координаты i -го города на плоскости. Координаты всех городов различны. В следующей, $N + 2$ -ой строке, задано число M . Далее в M строках записаны пары чисел $U_j V_j$, означающие, что существует дорога между городами U_j и V_j . Дороги не имеют общих точек вне городов. Длина дороги между городами A и B задаётся обычным евклидовым расстоянием между точками плоскости: $\rho(A, B) = \sqrt{(X_B - X_A)^2 + (Y_B - Y_A)^2}$.

Формат выходных данных

Выведите одно число — максимальное расстояние между двумя городами с точностью по крайней мере 10^{-6} . Если существует пара городов, между которыми нет пути из дорог, выведите -1 .

Примеры

maxdist.in	maxdist.out
2 0 0 1 1 1 1 2	1.414213
2 0 0 1 1 0	-1
5 0 0 0 2 2 0 2 2 1 1 6 1 2 2 4 4 3 3 1 2 5 4 5	4

Задача J. Треугольник и точка

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

Заданы прямоугольные координаты $x_1, y_1; x_2, y_2; x_3, y_3$ вершин треугольника и координаты точки x, y . Определить, находится ли точка в треугольнике. Погрешностями вычислений пренебречь.

Формат входных данных

Первая строка входного файла содержит N - количество тестов. Далее следует $2N$ строк, первая из которых содержит 6 действительных чисел разделённых пробелами - координаты вершин треугольника. А вторая - 2 действительных числа - координаты точки.

Формат выходных данных

Должен содержать N строк в которых для каждого из N тестов вывести YES если точка внутри треугольника (или на границе) и NO в противном случае.

Примеры

trianglep.in	trianglep.out
6	YES
1 4 2 3 0 4	YES
0 4	YES
4 2 2 0 0 4	NO
2 3	NO
2 4 1 3 5 0	NO
2 4	
4 4 3 3 1 4	
5 4	
2 3 0 4 5 0	
5 4	
2 4 4 3 5 3	
0 1	

Задача K. Две окружности

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

Заданы две окружности на плоскости. Задача заключается в нахождении всех точек их пересечения.

Формат входных данных

Первая строка входного файла содержит число наборов входных данных K ($1 \leq K \leq 10000$). Каждый набор состоит из двух строчек, каждая из которых описывает окружность. Описание окружности задается в виде трех чисел x, y, r ($-100 \leq x, y \leq 100, 0 < r \leq 100$), разделенным пробелами. Все числа во входном файле целые.

Формат выходных данных

Для каждого из наборов необходимо вывести одно из нижеследующих сообщений:

- “There are no points!!!” — если точки пересечения отсутствуют.
- “There are only i of them...” — если окружности имеют в точности i точек пересечения. В этом случае следующие i строчек должны содержать координаты точек x'_j и y'_j . Точки требуется выводить в порядке возрастания (сначала с меньшими x , если значения x равны, то с меньшими y). Числа необходимо выводить не менее чем с двенадцатью точными знаками после запятой.
- “I can’t count them - too many points :(" — если точек пересечения бесконечно много.

Все сообщения необходимо выводить без кавычек. Разделяйте вывод для различных наборов пустой строкой.

Примеры

circles.in	circles.out
2	There are only 1 of them....
0 0 2	2.000000000000 0.000000000000
4 0 2	There are no points!!!
0 0 1	
100 100 1	

Задача L. Треугольники++

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

Пифагор измерил длины сторон трех треугольников и записал их на клочке бумаги в случайном порядке. Через некоторое время он попытался разобрать свои записи и не смог определить какие размеры какому треугольнику соответствуют, но вспомнил, что, по крайней мере, два треугольника были подобными.

Даны размеры 9 сторон трех треугольников, записанные в случайном порядке. Указать - из каких сторон можно составить два подобных треугольника.

Формат входных данных

Во входном файле содержатся девять целых чисел от 1 до 1000, разделенных пробелами - длины сторон треугольников.

Формат выходных данных

В выходной файл записать в первой строке 3 числа, соответствующих сторонам одного из подобных треугольников, во второй строке 3 числа, соответствующих сторонам другого подобного треугольника. Если есть несколько вариантов, вывести один (любой) из них.

Примеры

pifagor.in	pifagor.out
2 15 3 9 3 4 12 3 5	3 4 5 9 12 15
2 3 4 4 6 8 5 10 3	3 4 5 6 8 10
2 4 3 6 4 8 5 10 3	4 3 5 8 6 10
2 3 5 2 3 5 2 3 5	2 2 2 3 3 3
22 10 36 11 12 30 33 20 25	10 11 12 30 33 36

Задача M. Колпак

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

Определите максимальный размер шара, который можно спрятать под “колпаком” – круглым прямым конусом (основание является кругом, ось конуса перпендикулярна основанию).

Формат входных данных

Во входном файле в первой строке содержится два числа, разделенных пробелом - длина образующей конуса L ($1 \leq L \leq 100$) и диаметр основания D ($1 \leq D < 2 * L$).

Формат выходных данных

В выходной файл вывести одно число с 4 десятичными знаками - радиус шара максимального размера, который может поместиться под заданным “колпаком”.

Примеры

pack.in	pack.out
5.0 6.0	1.5000
50.0 50.0	14.4338

