

Не забывайте, что $\pi \approx 3,14159265358979323846$. В языке Паскаль это константа π . В языке Си это константа M_PI , объявленная в файле `math.h`.

Задача А. Углы треугольника

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

Дан треугольник. Определите величину самого большого из его углов.

Формат входного файла

Программа получает на вход координаты трех вершин треугольника (сначала координаты первой вершины, затем второй, затем третьей). Координаты — пара целых чисел, не превосходящих 10^4 по модулю.

Формат выходного файла

Выведите единственное число — величину самого большого угла треугольника в градусах с точностью не меньше 6 знаков после запятой.

Пример

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

Задача В. 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

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 съедена?

Формат входного файла

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

Формат выходного файла

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

Пример

<code>goat1.in</code>	<code>goat1.out</code>
10 6	95.091113

Задача С. 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

Козла пустили в огород. К несчастью для козла, его крепко-накрепко привязали к точке С крепчайшей альпинистской верёвкой длины L (то есть козёл может кушать травку, удалённую от точки С 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 А и В. Но, при большом старании, козёл может растянуть крепчайшую альпинистскую верёвку. Определите, на какую длину козлу придётся растянуть верёвку, чтобы добраться хотя бы до одного ананасика? А чтобы съесть все ананасы?

Формат входного файла

На входе находятся координаты точек А, В, С и длина верёвки L . Все числа целые, $L \geq 0$, все координаты не превосходят по модулю 10000. Числа разделены пробелами или переломами строк.

Формат выходного файла

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>goat2.in</code>	<code>goat2.out</code>
8 -6 8 6	1.00
0 0 7	3.00

Задача D. 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

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 пастьись на участке огорода максимальной площади, но при этом не заходил бы на участки соседей. Определите, в каком месте огорода нужно привязать козла.

Формат входного файла

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

Формат выходного файла

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

Пример

<code>goat3.in</code>	<code>goat3.out</code>
0 0	1.0 1.0
3 0	
0 4	

Задача E. Пусти козла в огород - 4

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

В огород пустили двух козлов, каждого привязав верёвкой к своему колышку. Каждый козёл движется по окружности, растягивая верёвку на максимальную длину. Найдите все точки, в которых они могут встретиться.

Формат входного файла

В первых двух строках находятся по три числа — координаты колышка, к которому привязан каждый козёл, и длина

верёвки, которой он привязан. Все числа — целые, не превосходящие по модулю 10000, длины веревок — положительные.

Формат выходного файла

В первой строке выходного файла выведите количество точек пересечения (0, 1, 2 или 3, если их количество бесконечно). В случае одной точки пересечения выведите во второй строке координаты этой точки. В случае двух точек пересечения выведите координаты двух точек в лексикографическом порядке (то есть, сначала выведите ту точку, у которой меньше координата x , а если координаты x равны, то ту точку, у которой меньше координата y). Все числа следует выводить с точностью не менее 6 знаков после запятой.

Примеры

goat4.in	goat4.out
3 4 5 11 4 2	0
3 4 5 9 4 2	2 7.7500 2.4387505 7.7500 5.5612495

Задача F. 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

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 входного файла

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 выходного файла

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

Пример

goat5.in	goat5.out
8	36.8060473804
1 1	
-2 0	
5 3	
-2 3	
8 3.10	
-2 -1	
-2 2	
8 4.10	

Задача G. Пересечение отрезков

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

Даны два отрезка: AB и CD . Определите, какое множество точек является пересечением этих отрезков.

Формат входного файла

Программа получает на вход восемь целых чисел, по абсолютной величине не превосходящих 10^4 — координаты точек A, B, C, D . Точки могут совпадать (в том числе могут совпадать и концы одного отрезка).

Формат выходного файла

Если указанные отрезки не пересекаются, то выведите строку "Empty". Если отрезки пересекаются 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.

Пример

segments.in	segments.out
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