

Задача А. Векторы

Имя входного файла: `vectors.in`

Имя выходного файла: `vectors.out`

Ограничение по времени: 2 секунды

Ограничение по памяти: 64 мегабайта

Даны два ненулевых вектора. Требуется вычислить:

- Длину первого и второго вектора (два числа)
- Вектор, образованный сложением данных двух векторов
- Скалярное и векторное произведения данных векторов
- Площадь треугольника, построенного из этих векторов
- Полярные углы, которые образуют данные радиус-векторы (в интервале $[0; 2\pi)$).
- Угол между заданными векторами в радианах (в интервале $[0; 2\pi)$).
- Часть плоскости, которую занимает сектор, образованный двумя данными векторами (предыдущее значение, делённое на 2π).

Формат входных данных

В двух строках входного файла заданы по четыре числа — координаты начала и конца первого вектора, затем второго. Все числа целые, по модулю не превосходящие 10000.

Формат выходных данных

В каждой строке выходного файла — ответ на соответствующий пункт задачи с точностью не менее 10^{-6} .

Примеры

<code>vectors.in</code>	<code>vectors.out</code>
5 1 2 6 1 1 7 8	5.830951895 9.219544457 3.000000000 12.000000000 17.000000000 -51.000000000 25.500000000 2.111215827 0.862170055 5.034139535 0.801208191

Задача С. В каком ухе мяукает Пушок

Имя входного файла:	<code>buzz.in</code>
Имя выходного файла:	<code>buzz.out</code>
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

Кирилл знает, что его котик Пушок охотится на новогоднюю ёлочку и хочет её опрокинуть. Кирилл зашел в комнату, в котором находится Пушок и ёлочка, чтобы поймать котика и выгнать его из комнаты. Пушок очень хитёр — он спрятался в комнате, но его выдаёт тихое мяукание. Кирилл находится в точке $A(x_a; y_a)$ и, глядя прямо на ёлочку, стоящую в точке $B(x_b; y_b)$ задает вопрос: «В каком ухе у меня раздаётся мяукание?». Пушок спрятался и мяукает в точке $C(x_c; y_c)$.

Помогите Кириллу спасти ёлочку — понять, в каком ухе у него раздаётся мяукание.

Формат входных данных

Программа получает на вход координаты точек A , B и C . Входные данные являются целыми числами, по модулю не превышающими 1 000.

Формат выходных данных

Выведите слово `LEFT`, если у Кирилла раздается мяукание в левом ухе, `RIGHT` — если в правом, `BOTH` — если мяукание и в левом, и в правом одинаково.

Примеры

<code>buzz.in</code>	<code>buzz.out</code>
1 0 3 0 0 0	BOTH

Задача D. Угол между векторами

Имя входного файла: angle2.in
Имя выходного файла: angle2.out
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Формат входных данных

Четыре целых числа, по модулю не превышающие 10^4 — координаты двух ненулевых векторов.

Формат выходных данных

Одно число — величина неориентированного угла между ними с точностью до пятого знака после запятой из интервала $[0, \pi]$.

Примеры

angle2.in	angle2.out
2 1 3 5	0.56672921752350635

Задача Е. Полярный угол

Имя входного файла: `angle1.in`
Имя выходного файла: `angle1.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Формат входных данных

Два числа — координаты точки ($|x|, |y| \leq 10^4$), не совпадающей с началом координат.

Формат выходных данных

Одно число — величина её полярного угла в радианах из интервала $[0, 2\pi)$. Все числа следует выводить с точностью не менее 6 знаков после запятой.

Примеры

<code>angle1.in</code>	<code>angle1.out</code>
2 3	0.98279372324732906796

Задача F. Принадлежность точки промежутку

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

Определите, принадлежит ли точка C заданной прямой, лучу и отрезку, образованными точками A и B .

Формат входных данных

В первой строке входного файла даны два целых числа — координаты точки C . Во двух следующих строках в таком же формате заданы точки A и B ($A \neq B$).

Все числа во входном файле по модулю не превосходят 10 000.

Формат выходных данных

В первой строке выведите «YES», если точка C принадлежит прямой AB , и «NO» в противном случае. Во второй и третьей строках аналогично выведите ответы для луча AB (A — начало луча) и отрезка AB .

Примеры

<code>point.in</code>	<code>point.out</code>
1 6	YES
3 7	NO
5 8	NO

Задача G. Штрафы

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

Новый градоначальник города Глупова решил с целью пополнения бюджета и экономии горючего провести кампанию борьбы с левым уклоном и левыми рейсами. Для этого он запретил водителям выполнять левые повороты, установив штраф за каждый поворот налево в размере одного миллиона (разворот поворотом налево не считается).

От тяжелого прошлого Глупову достались улицы, которые могут пересекаться под любыми углами. Градоначальник приказал установить компьютерную систему тотальной слежки, которая следит за каждым автомобилем, записывая его координаты каждый раз, когда тот меняет направление движения (включая начальную и конечную точки пути).

Требуется написать программу, вычисляющую по записанной последовательности координат автомобиля штраф, который должен быть взыскан с водителя.

Формат входных данных

В первой строке вводится целое число N - количество записанных пар координат ($1 \leq N \leq 1000$). В каждой из следующих N строк записана очередная из этих пар (вещественные числа).

Формат выходных данных

Выведите суммарный штраф водителя в миллионах.

Примеры

стандартный ввод	стандартный вывод
4	1
0 0	
1 0	
1 1	
2 1	

Задача Н. Расстояния от точки

Имя входного файла: `distance1.in`
Имя выходного файла: `distance1.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Даны три точки A , B и C . Необходимо подсчитать расстояния от точки C до прямой, луча и отрезка, образованного точками A и B .

Формат входных данных

В первой строке входного файла даны два целых числа — координаты точки A . Во двух следующих строках в таком же формате заданы точки B и C . Все числа не превосходят по модулю 10 000.

Формат выходных данных

В первой строке выходного файла выведите одно вещественное число — расстояние от точки C до прямой. В следующих двух строках выведите соответственно расстояния до луча и до отрезка. Все числа выводить с точностью не менее 10^{-6} . Луч строится по направлению от точки A к точке B .

Примеры

<code>distance1.in</code>	<code>distance1.out</code>
1 1	1.0000000000
2 1	1.0000000000
3 0	1.4142135624

Задача I. Пересечение двух отрезков

Имя входного файла: `intersec2.in`

Имя выходного файла: `intersec2.out`

Ограничение по времени: 2 секунды

Ограничение по памяти: 64 мегабайта

Необходимо проверить, пересекаются ли два отрезка.

Формат входных данных

В единственной строке входного файла даны восемь целых чисел — координаты концов сначала первого, затем второго отрезков. Все числа не превышают 10 000.

Формат выходных данных

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

Примеры

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

Задача J. Пусти козла в огород - 1

Имя входного файла: `goat1.in`
Имя выходного файла: `goat1.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

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 точек, 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.

Примеры

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

Задача К. Пусти козла в огород - 3

Имя входного файла: `goat3.in`
Имя выходного файла: `goat3.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

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

В первой строке входного файла содержатся координаты точек A и B , разделённые пробелами. На второй строке записаны координаты точки C и длина верёвки L , разделённые пробелами. Все числа целые, $L \geq 0$, все координаты не превосходят по модулю 10 000.

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.

Примеры

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

Задача L. Пусти козла в огород - 4

Имя входного файла: **goat4.in**
Имя выходного файла: **goat4.out**
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

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.

Примеры

goat4.in	goat4.out
0 0	1.00000000 1.00000000
3 0	
0 4	

Задача М. Пусти козла в огород - 5

Имя входного файла: `goat5.in`
Имя выходного файла: `goat5.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

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 услышал 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 N. Верни козла в огород

Имя входного файла:	goat-go-home.in
Имя выходного файла:	goat-go-home.out
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Любимому козлу Петра Васильевича наконец-то удалось сбежать от своего хозяина! После длительных поисков Пётр Васильевич вышел на берег большого озера, где и обнаружил беглеца. Правда, на небольшом островке в центре озера.

К счастью, у соседа Василия Петровича в сарае нашлась старая моторная лодка. Пётр Васильевич незамедлительно притащил её на берег, завёл мотор и поплыл прямо по направлению к козлу. Однако в середине пути мотор неожиданно заглох.

С помощью лома, кувалды, подручных материалов и отборных слов из последнего тура игры «Завалинка» Пётр Васильевич наконец заставил мотор работать. Однако спустя некоторое время он с удивлением обнаружил, что теперь соседская лодка в каждый момент времени может плыть только строго в одном из n фиксированных направлений.

Итак, сейчас лодка вместе с Петром Васильевичем расположена в точке (x_s, y_s) . Козёл неподвижно стоит в точке (x_t, y_t) . Считайте, что на озере штиль (ветра нет). Любое из доступных направлений можно использовать сколько угодно раз на протяжении любого вещественного неотрицательного промежутка времени. Одновременно может быть использовано только одно из направлений, и, кроме них, на лодку не действуют никакие другие силы.

Вам поручено определить, существует ли стратегия использования ветров, которая позволит Петру Васильевичу добраться до козла.

Формат входных данных

В первой строке входных данных находятся два целых числа x_s и y_s — текущие координаты лодки. Во второй строке записаны целые числа x_t и y_t , задающие местоположение козла Петра Васильевича ($|x_s|, |y_s|, |x_t|, |y_t| \leq 10^9$).

Третья строка содержит единственное число n — количество направлений, в которых может двигаться лодка ($0 \leq n \leq 10$).

Следующие n строк содержат пары целых чисел x_i и y_i . Соответствующее направление задаётся вектором (x_i, y_i) . Мощность этого направления (то есть расстояние, которое преодолеет лодка за единицу времени) определяется как неотрицательное вещественное число, равное $\sqrt{x_i^2 + y_i^2}$ ($|x_i|, |y_i| \leq 10^9$).

Формат выходных данных

Если Пётр Васильевич может доплыть до своего козла, то выведите «YES» (без кавычек). В противном случае выведите «NO» (без кавычек).

Примеры

goat-go-home.in	goat-go-home.out
1 1 2 2 2 1 0 0 2	YES
5 6 7 8 2 2 0 2 1	NO

Задача О. Площадь многоугольника

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

На плоскости заданы координаты вершин многоугольника в порядке их обхода. Многоугольник не обязательно выпуклый, но не содержит самопересечений. Требуется найти его площадь.

Формат входных данных

Сначала записано число N — количество вершин многоугольника ($3 \leq N \leq 100$), затем N пар вещественных чисел, задающих координаты его вершин x_i, y_i ($-1000 \leq x_i, y_i \leq 1000$).

Формат выходных данных

Выведите площадь многоугольника не меньше, чем с 3 знаками после десятичной точки.

Примеры

area.in	area.out
4	11.0
0 0	
0 2	
4 3.5	
4 0	

Замечание

Если выводить вещественные числа как `print(x)`, то иногда они будут странно отформатированы, например, $10^{-6} = \frac{1}{1000000}$ будет выведено как `1e-6`.

Поэтому числа с заданной точностью следует переводить в строку так:

```
x = 1.34
print("{:.6f}".format(x)) # строка "1.340000"
```

Для того, чтобы избежать проблем с погрешностью вашего ответа, если условие это позволяет, следует выводить числа с максимально возможной точностью, для типа `float` в Питоне это 16 десятичных знаков:

```
print("{:.16f}".format(x))
```

Задача Р. Площадь многоугольника

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

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.

Координаты целые по модулю не превосходят 10 000.

Формат выходных данных

Одно число — величина площади приведённого многоугольника с абсолютной точностью.

Примеры

area.in	area.out
3 1 0 0 1 1 1	0.5