

Задача А. Базовые операции над вектором

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

Даны две несовпадающих точки на плоскости, требуется вычислить:

- Вектор с началом в первой и концом во второй точках
- Соответствующий ему нормированный вектор
- Вектор, сонаправленный первому и имеющий заданную длину
- Вектор, полученный путем поворота первого вектора на 90° по часовой стрелке
- Вектор, полученный путем поворота первого вектора на 90° против часовой стрелки

Формат входных данных

В первых двух строках даны по два целых числа, — координаты заданных точек. В третьей строке записано натуральное число, — длина, которую должен иметь построенный в третьем пункте задачи вектор.

Все числа во входном файле по модулю не превосходят 1000.

Формат выходных данных

В отдельных строках выходного файла нужно вывести координаты векторов, соответствующих каждому из пунктов задачи, с точностью до 10^{-4} .

Примеры

<code>basicvector.in</code>	<code>basicvector.out</code>
3 6	0 2
3 8	0.000000 1.000000
5	0.000000 5.000000
	2 0
	-2 0

Задача В. 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
- Угол между заданными векторами
- Часть плоскости, которую занимает сектор, образованный двумя данными векторами.

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	5.830951895 9.219544457
1 1 7 8	3.000000000 12.000000000
	17.000000000 -51.000000000
	25.500000000
	2.111215827 0.862170055
	5.034139535
	0.801208191

Задача С. Прямая

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

- Построить направляющий вектор этой прямой и вывести его координаты
- Найти коэффициенты A , B и C нормального уравнения прямой
- Найти координаты любого вектора, перпендикулярного данной прямой
- Найти коэффициенты A_1 , B_1 , C_1 прямой, перпендикулярной данной
- Построить прямые, параллельные данной, и находящиеся на заданном расстоянии от неё (вывести коэффициенты нормального уравнения A_2 , B_2 , C_2)

Формат входных данных

В первой строке входного файла находятся четыре числа — координаты двух различных точек на прямой. Во второй строке находится требуемое расстояние для построения параллельной прямой.

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

Формат выходных данных

В каждой строке выходного файла выведите ответ на соответствующий пункт задачи с точностью до 10^{-6} .

Примеры

<code>line.in</code>	<code>line.out</code>
1 2 3 1	2.000000000 -1.000000000
3	-1.000000000 -2.000000000 5.000000000
	-1.000000000 -2.000000000
	2.000000000 -1.000000000 0.000000000
	-1.000000000 -2.000000000
	11.708203932
	-1.000000000 -2.000000000
	-1.708203932

Задача D. Уравнение прямой II

Имя входного файла: `line2.in`
Имя выходного файла: `line2.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Формат входных данных

Четыре числа — координаты точки на прямой и координаты вектора нормали к этой прямой.

Формат выходных данных

Три числа — коэффициенты A , B и C нормального уравнения этой прямой.

Примеры

<code>line2.in</code>	<code>line2.out</code>
1 2 3 1	3 1 -5

Задача Е. Две прямые

Имя входного файла: `lines2.in`
Имя выходного файла: `lines2.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

Даны коэффициенты A_1, B_1, C_1 уравнения первой прямой и коэффициенты A_2, B_2, C_2 уравнения второй прямой. Требуется:

- Построить направляющие векторы для обеих прямых (вывести координаты)
- Найти точку пересечения двух прямых или вычислить расстояние между ними, если они параллельны

Формат входных данных

В первой строке входного файла находятся три числа — коэффициенты нормального уравнения для первой прямой. Во второй строке — коэффициенты для второй прямой.

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

Формат выходных данных

В каждой строке выходного файла выведите ответ на соответствующий пункт задачи с точностью до 10^{-6} .

Примеры

<code>lines2.in</code>	<code>lines2.out</code>
<code>1 1 -1</code>	<code>1.000000000 -1.000000000</code>
<code>1 -1 0</code>	<code>-1.000000000 -1.000000000</code>
	<code>0.500000000 0.500000000</code>

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. Пушок и две ёлочки

Имя входного файла: `distance1.in`
Имя выходного файла: `distance1.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайта

Кирилл решил купить вторую ёлочку, чтобы, если Пушок повалит одну из них, то останется вторая, и Кириллу будет не так обидно. Но Пушок запланировал атаку на обе ёлочки. Для более эффективной атаки ёлочки и чтобы быстрее прятаться от Кирилла, Пушку нужно рассчитать расстояния от себя (точки C) до прямой, луча и отрезка, образованного точками A и B , в которых находятся ёлочки.

Формат входных данных

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

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

Формат выходных данных

В первой строке выходного файла выведите одно вещественное число — расстояние от точки C с Пушком до прямой. В следующих двух строках выведите соответственно расстояния до луча AB (A — начало луча) и до отрезка AB . 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>
3 0	1.000000000
1 1	1.000000000
2 1	1.414213562

Задача Н. Расстояние между отрезками

Имя входного файла: distance5.in
Имя выходного файла: distance5.out
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Найдите расстояние между двумя отрезками.

Формат входных данных

Восемь целых чисел — координаты четырёх концов двух отрезков.

Формат выходных данных

Одно число — расстояние между отрезками с точностью не менее 10^{-6} .

Примеры

distance5.in	distance5.out
1 1 2 2	0.7071067812
2 1 3 0	

Задача I. Пусти козла в огород - 8

Имя входного файла: `goat8.in`
Имя выходного файла: `goat8.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 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 входных данных

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 выходных данных

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

<code>goat8.in</code>	<code>goat8.out</code>
3 4 5 11 4 2	0
3 4 5 9 4 2	2 7.7500000000 2.4387505004 7.7500000000 5.5612494996

Задача J. Точка в многоугольнике

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

Формат входных данных

В первой строке содержится три числа — N ($3 \leq N \leq 100\,000$) 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.

Примеры

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

Задача К. Прогулки Сени

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

Сеня очень любит гулять. Вот и сегодня он пошёл в парк на прогулку.

Сеня решил запомнить свой путь на будущее, вдруг ему на нем понравится какое-нибудь место. Для этого он взял с собой карту, и ставил там точки каждый раз, когда поворачивал или встречал что-то интересное. В результате его пути на карте образовалась ломаная.

Придя домой, он нашёл что один его ботинок стёрся чуть сильнее, чем другой. Сеня предположил, что это из-за того, что поворотов в одну сторону было больше.

Проверьте Сенину догадку, и выведите для данной ломаной, сколько раз она поворачивает налево, а сколько раз — направо.

Учтите, что иногда Сеня возвращается назад или случайно ставит точку там, где поворота не было. Оба таких случая Сеня за поворот не считает.

Формат входных данных

В первой строке дано число N ($3 \leq N \leq 10^6$) — количество точек на пути Сени.

Каждая из последующих N строк содержит координаты точки x_i, y_i ($-10\,000 \leq x_i, y_i \leq 10\,000$). Две соседние точки не являются равными.

Точки даны в порядке обхода их Сеней.

Формат выходных данных

Выведите в одной строке два числа: сколько раз Сеня повернул налево, а сколько раз — направо.

Примеры

<code>walk.in</code>	<code>walk.out</code>
3 0 1 1 0 1 1	1 0
7 0 0 1 1 2 1 3 1 3 3 3 0 0 0	1 2

Замечание

TL для Python 5 секунд

Задача L. Касательная к окружности

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

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 ($1 \leq r \leq 10^4$).

В третьей строке находятся два числа — координаты точки P_1 , из которой нужно провести касательные. Координаты — целые числа не превышающие по модулю 10^4 .

Формат выходных данных

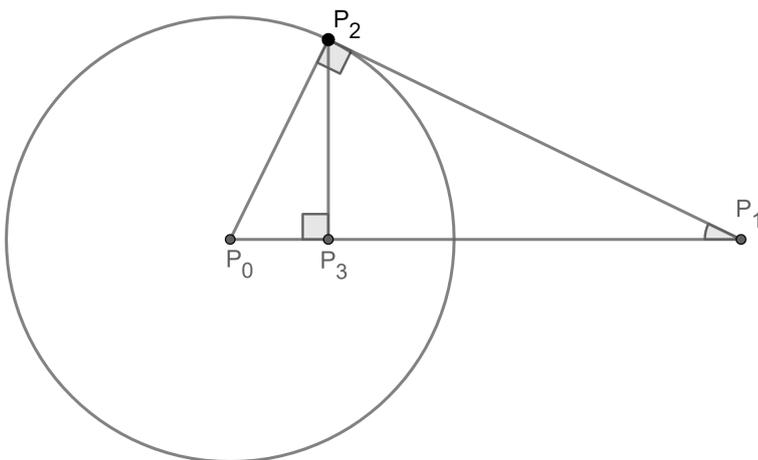
В первой строке выходного файла выведите количество точек касания (0, 1 или 2).

- В случае одной точки касания выведете во второй строке координаты этой точки.
- В случае двух точек касания выведете во второй строке координаты точки P_3 — проекции точки касания на прямую P_0P_1 (см. картинку). В третьей строке — длины векторов $\overrightarrow{P_1P_3}$ и $\overrightarrow{P_3P_2}$. В последних двух строках должны находиться координаты точек касания.

Примеры

<code>tangent.in</code>	<code>tangent.out</code>
2 2	2
2	2.0000000000 3.3333333333
2 5	1.6666666667 1.4907119850
	0.5092880150 3.3333333333
	3.4907119850 3.3333333333

Замечание



Задача М. Ловушка для Слонопотама

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

Пятачок и Винни-Пух каждое утро ходят пить чай в гости к Кролику. Естественно, самым коротким путем.

К сожалению, однажды Винни-Пуху пришла в голову идея вырыть ловушку для Слонопотама. Самое обидное, что они с Пятачком ее даже вырыли. Поэтому теперь каждое утро, идя в гости к Кролику, они боятся в нее провалиться.

Напишите программу, которая посчитает длину самого короткого безопасного пути от домика Винни-Пуха до домика Кролика.

Ловушка для Слонопотама представляет собой яму абсолютно круглой формы. Путь является безопасным, если он не проходит по ловушке (но может проходить по ее границе).

Формат входных данных

Во входном файле записаны сначала координаты домика Винни-Пуха: X_B, Y_B , затем — координаты домика Кролика: X_R, Y_R , а затем — координаты центра и радиус ловушки: X_T, Y_T, R_T . Все координаты — целые числа из диапазона от -32000 до 32000 . Радиус ловушки — натуральное число, не превышающее 32000 .

Домики Винни-Пуха и Кролика не могут находиться внутри ловушки, но могут находиться на ее границе.

Формат выходных данных

Выведите в выходной файл одно число — длину самого короткого безопасного пути от домика Винни-Пуха до домика Кролика с точностью не менее 4 знака после запятой.

Примеры

piglet.in	piglet.out
0 0 0 1 10 10 1	1.000000
5 0 0 5 0 0 5	7.853982
-5 0 5 0 0 0 3	11.861007

Задача N. Река

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

Каждый день в Монастыре Светлой Луны проходит утренний обряд, являющийся важнейшим элементом жизненного пути каждого монаха. Во время этого обряда один из монахов должен выйти из Монастыря, пройти к реке Мацанг (истоку Брахмапутры) и, набрав там воды, принести ее в Храм Солнца Дхармы, расположенный неподалеку от Монастыря. При этом обряд должен быть закончен ровно к рассвету. Для того чтобы достичь такой точности, монахам необходимо знать длину кратчайшего пути от Монастыря до Храма, проходящего через реку Мацанг. Вы призваны помочь Монастырю Светлой Луны рассчитать эту самую длину. Поскольку река близ Монастыря не имеет сколько-нибудь значительных изгибов, вы можете считать ее идеально прямой. Кроме того, можно пренебречь ее шириной. Однако необходимо помнить, что Мацанг — река горная и опасная, а поэтому перейти ее вброд абсолютно невозможно! При этом, естественно, ни Монастырь, ни Храм не находятся в реке.

Формат входных данных

В первой строке записаны координаты Монастыря, во второй — координаты Храма. В третьей строке записаны числа k, b — коэффициенты, задающие уравнение реки ($y = kx + b$). Все числа кроме k — целые и не превосходят по модулю 10000. Коэффициент k — действительное число, не превосходящее по модулю 100.

Формат выходных данных

Выведите одно действительное число, округленное до трех знаков после запятой — длину кратчайшего пути. В том случае, если Монах не может пройти к Храму, необходимо вывести "No solution."

Примеры

<code>river.in</code>	<code>river.out</code>
0 10 20 10 0 0	28.284
0 10 20 -10 0 0	No solution.

Задача О. Велоклуб

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

Павел Денисович уже который год привозит с собой в ЛКШ велосипед. Сегодня он решил провести велоклуб и проложил маршрут по дороге, ведущей строго на север. По ней он и поведёт за собой колонну из трёх-четырёх школьников во время велоклуба.

Но метеорадар говорит о том, что с востока движется странная грозовая туча в форме квадрата, наклонённого под углом 45° . Это значит, что велоклуб находится под угрозой срыва.

Скажите, попадёт ли колонна под дождь, если она будет ехать вверх по дороге из точки $(0, -p_b)$ со скоростью V_b , а центр самой тучи с диагональю $2r$ движется из точки $(-p_c, 0)$ в сторону дороги, перпендикулярно ей со скоростью V_c . Колонну считайте движущейся точкой. Если колонна попадает на край тучи, то она всё равно попадает под дождь. Туча движется медленнее, чем велоклуб.

Формат входных данных

В первой строке через пробел указаны два числа: p_b и V_b ($0 \leq p_b \leq 100$, $0 < V_b \leq 100$).

Во второй строке через пробел указаны три числа p_c , r_c и V_c ($0 \leq p_c, r_c \leq 100$, $0 < V_c \leq 100$).

Формат выходных данных

Если Велоклуб попадёт под дождь, выведите «YES». Иначе выведите «NO». Слова нужно вывести без кавычек.

Примеры

diamond-chase.in	diamond-chase.out
5 2	YES
2 3 2	

Замечание

В первом тесте из условия в момент времени $t = 1$ велогруппа будет находится в точке $(0, 3)$. Там же будет находиться нижний край грозовой тучи, значит колонна попадёт под дождь.

