

Задача А. Длина вектора

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

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

Формат входного файла

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

Формат выходного файла

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
2.7 0.9 1.4	3.17
1 2 2	3.00
1.8 1.1 0.4	2.15

Задача В. Сумма векторов

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

Вычислить сумму векторов $\vec{c} = \vec{a} + \vec{b}$ заданных своими координатами.

Формат входного файла

Первая строка входного файла содержит три вещественных числа – координаты a_x, a_y, a_z , разделённые пробелами. Вторая строка – координаты b_x, b_y, b_z .

Формат выходного файла

Первая строка выходного файла должна содержать координаты \vec{c} – вектора суммы (каждая координата с точностью до 2 знаков после десятичной точки).

Примеры

<code>sumvec.in</code>	<code>sumvec.out</code>
1 2 3 4 5 6	5.00 7.00 9.00
5.57 0.56 1.00 4.46 4.50 3.92	10.03 5.06 4.92

Задача С. Параллельность

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

На плоскости заданы две прямых. Определить, параллельны ли прямые?

Прямые заданы коэффициентами уравнения $Ax + By + C = 0$.

Формат входного файла

Входной файл содержит несколько (до 1000) тестов. Каждый тест занимает 2 строки, в каждой по 3 числа: A_1, B_1, C_1 , и A_2, B_2, C_2 . Все числа целые и не превосходят

по модулю 10^4 . Окончание тестов – “0 0 0” на отдельной строке.

Формат выходного файла

Для каждого теста выведите “YES” если прямые параллельны или совпадают, и “NO” в противном случае.

Примеры

<code>par.in</code>	<code>par.out</code>
0 1 5	YES
0 2 6	NO
1 2 3	YES
-1 2 6	
1 2 3	
2 4 0	
0 0 0	

Задача D. Скалярное произведение

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

Скалярное произведение – операция над двумя векторами, результатом которой является *скаляр* (число), не зависящее от системы координат и характеризующее длины векторов-сомножителей и угол между ними.

Обычно используется одно из следующих обозначений: $\langle \mathbf{a}, \mathbf{b} \rangle$, (\mathbf{a}, \mathbf{b}) , $\mathbf{a} \cdot \mathbf{b}$, или (обозначение Дирака, часто применяемое в квантовой механике для векторов состояния): $\langle a|b \rangle$.

В трёхмерном евклидовом пространстве $\vec{x} = (x_1, x_2, x_3)$ скалярное произведение $\langle \vec{x}, \vec{y} \rangle = x_1 y_1 + x_2 y_2 + x_3 y_3$. Аналогичное утверждение верно для евклидова пространства любой размерности (в сумму тогда входит количество членов, равное размерности пространства).

Нужно посчитать скалярное произведение 2-х векторов в N -мерном пространстве.

Формат входного файла

В первой строке входного файла задано число N ($1 \leq N \leq 100$) – размерность пространства.

Следующие две строки содержат по N чисел (x_1, x_2, \dots, x_N) – вектора. Координаты – вещественные числа заданные с точностью до 10^{-4} .

Формат выходного файла

В первой строке входного файла выведите вещественное число – скалярное произведение с точностью до 3-х знаков после запятой.

Примеры

<code>scalar.in</code>	<code>scalar.out</code>
3 2 3 9 3 6 6	78.000
2 2.1 3.3 -6.6 4.2	0.000

Задача Е. Тангенс

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

Посчитайте тангенс угла заданного в градусах.

Формат входного файла

Первая строка содержит количество наборов исходных данных T ($0 \leq T \leq 100$). Далее T строк — углы в градусах a (целое число) ($0 \leq a \leq 10000$).

Формат выходного файла

Для каждого из углов необходимо вывести значение тангенса с 3-мя знаками после запятой.

Примеры

tang.in	tang.out
6	0.000
0	0.577
30	1.000
45	1.732
60	57.290
89	0.017
361	

Задача F. Расстояние

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

На плоскости заданы точка и прямая. Требуется найти расстояние между ними.

Прямая задана уравнением $Ax + By + C = 0$. Точка задана координатами $(x_0; y_0)$.

Формат входного файла

Входной файл содержит несколько (до 1000) тестов. Каждый тест записан в отдельной строке и содержит 5 чисел A, B, C, x_0, y_0 разделённых пробелами. Все числа целые и не превосходят по модулю 10000. Окончание тестов — "0 0 0" на отдельной строке.

Формат выходного файла

Для каждого теста в выходной файл одно число — расстояние с точностью до 3-х знаков после точки.

Примеры

dist.in	dist.out
0 1 2 2 3	5.000
4 0 0 6 4	6.000
2 1 3 7 6	10.286
0 0 0	

Задача G. Картинная галерея

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

В картинной галерее, имеющей форму N -угольника, расположено M люстр, которые мы будем считать точечными источниками света. Точка стены галереи называется освещенной, если из нее видна хотя бы

одна из люстр. Неосвещенным участком будем называть максимальное связанное множество точек стены галереи, ни одна из которых не освещена (участок может содержать углы галереи). Напишите программу, определяющую все неосвещенные участки.

Рассмотрим решение задачи, если источник света один. Разобьем все стороны N -угольника на отрезки так, чтобы каждый отрезок был либо целиком освещен, либо целиком неосвещен. Для этого проведем из источника света лучи через те вершины N -угольника, угол при которых больше 180 градусов, и найдем ближайшие точки пересечения этих лучей со сторонами N -угольника (касания не рассматриваются). Эти точки и дадут нам требуемое разбиение. Теперь для каждого отрезка надо проверить, освещен он или нет. Проведем луч из источника света через середину проверяемого отрезка. Найдем точки пересечения этого луча со сторонами N -угольника и среди них отыщем наименее удаленную от источника света. Если это середина нашего отрезка, то отрезок освещен, если нет, то та сторона, которой принадлежит эта точка, закрывает проверяемый отрезок от источника света. Проведем эту процедуру для остальных источников света, разбивая имеющиеся неосвещенные отрезки на более мелкие (для освещенных отрезков это бессмысленно). При выводе соседние неосвещенные участки надо объединить.

Формат входного файла

Первая строка входного файла содержит два целых числа N и M ($1 \leq N, M \leq 30$). В каждой из следующих N строк записаны координаты очередного угла галереи. Углы перечислены в порядке обхода стены по часовой стрелке. Далее идут M строк, каждая из которых содержит координаты очередной из люстр. Все координаты являются вещественными числами и разделяются пробелом.

Формат выходного файла

В первую строку выходного файла выведите количество неосвещенных участков S . Каждая из следующих S строк должна содержать описание очередного из участков в виде тройки чисел, разделенных пробелом. Первые два числа определяют координаты начальной точки участка, третье — его длину. (Участок должен продолжаться на указанную длину в направлении обхода стены по часовой стрелке. Никакие два участка не должны иметь общих точек.) Числа, определяющие участок, должны быть выведены не менее чем с 3 верными значащими цифрами.

Примеры

gallery.in	gallery.out
5 1 0 0 0 5 4 5 2 3 5 0 3.0 1.0	1 1 5 5.82843
3 2 0 0 0 1 1 0 0.4 0.4 0.1 0.5	0
3 0 0 0 0 1 1 0	1 0.0000 0.0000 3.4142
6 1 0 0 -2 1 -1 1 0 2 1 1 2 1 0 1.5	2 -1.5000 0.7500 1.5590 1.0000 1.0000 1.5590