

Задача А. Река

Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

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

| river.in | river.out |
|-----------------------|--------------|
| 0 10 20 10 0 0 | 28.284 |
| 0 10 20 -10 0 0 | No solution. |

Задача В. Деревья

Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайт

В вашем саду растет несколько деревьев, которые необходимо поливать. Так уж получилось, что каждое из растений имеет форму круга, причем никакие два из них не пересекаются. Вы собираетесь купить две поливальные машины. Каждая из машин поливает всю землю внутри круга радиуса R . Одна из машин будет включаться по утрам, а другая по ночам. Чтобы растение не засохло, нужно чтобы либо утром, либо ночью оно целиком поливалось какой-то машиной. Т.е. окружность, представляющая каждое из деревьев, должна целиком содержаться в одной или обеих окружностях, представляющих поливальные машины.

Даны координаты и радиус каждого из деревьев. Найдите наименьший радиус R , для которого можно расположить обе поливальные машины так, чтобы ни одно дерево не засохло. Поливальные машины можно ставить в любую точку плоскости, даже внутри дерева.

Формат входных данных

В первой строчке дано $1 \leq N \leq 3$ — количество деревьев. Далее следует N строк из целых чисел x_i, y_i, r_i ($1 \leq x_i, y_i, r_i \leq 1000$) — координаты и радиус деревьев.

Формат выходных данных

Выведите одно число R — искомый радиус с точностью не менее 6 знаков после запятой.

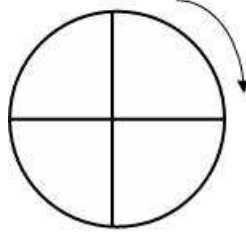
Примеры

| <code>circles.in</code> | <code>circles.out</code> |
|------------------------------------|--------------------------|
| 3 20 10 2 20 20 2 40 10 3 | 7.000000 |
| 3 20 10 3 30 10 3 40 10 3 | 8.000000 |

Задача С. Вращающаяся дверь

Ограничение по времени: 2 секунды
Ограничение по памяти: 64 Мб

В новом торговом центре проектируется вращающаяся дверь, которая на виде сверху представляет собой круг, разделенный на 4 равные части. Владельцы торгового центра знают, что самый большой товар, который у них будет продаваться, упаковывается в прямоугольные коробки шириной a , длиной b и высотой c . При этом транспортировать такую коробку можно только поставив ее на одну из граней. Определите, какой минимальный радиус должен быть у вращающейся двери, чтобы покупатель мог унести такой товар (высота двери может быть сколь угодно большой). Считается, что покупатель при попытке унести коробку действует самым очевидным для себя способом — размещает коробку так, чтобы угол один из углов коробки оказался в углу между створками двери. Размерами покупателя можно пренебречь



Формат входных данных

Натуральные числа a , b и c , не превосходящие 10000

Формат выходных данных

Выведите минимально необходимый радиус двери с точностью не менее 5 знаков после запятой.

Примеры

| door.in | door.out |
|---------|-----------|
| 1 1 1 | 1.4142136 |

Задача D. Приземление

Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Космический корабль приземляется на прямоугольную площадку космодрома. Правда, пилот не слишком опытен, поэтому возможны некоторые неточности. . .

Площадка представляет собой прямоугольник со сторонами, параллельными осям координат. Корабль (вид сверху) — выпуклый многоугольник.

Точка находится в безопасности, если расстояние от нее до корабля не меньше d . Автоматические системы не дадут посадить корабль так, чтобы хотя одна точка вне площадки не оказалась в безопасности. Также эти системы не разрешат вращение корабля в плоскости площадки, однако разрешат его параллельный перенос. Гарантируется, что корабль можно посадить.

Вычислите площадь той части площадки космодрома, все точки которой будут в безопасности при любом случае посадки, не нарушающем указанные выше правила.

Формат входных данных

В первой строке входного файла находятся 4 целых числа: N (количество вершин многоугольника, описывающего корабль), W , H (размеры площадки) и d (безопасное расстояние). $3 \leq N \leq 100$, $0 \leq W, H \leq 10^4$. Затем следует N пар целых чисел $X_i Y_i$ через пробел, по одной паре в каждой строке ($0 \leq X_i \leq W$, $0 \leq Y_i \leq H$) — координаты планируемого положения корабля, заданные в порядке обхода. Последовательные точки в порядке обхода границы многоугольника могут лежать на одной прямой и даже совпадать.

Формат выходных данных

Выходной файл должен содержать одно вещественное число с не менее, чем 6 точными знаками после запятой — площадь космодрома, на которой находиться будет безопасно.

Пример

| landing.in | landing.out |
|--|---------------|
| 3 100 100 1 10 10 20 10 10 20 | 56.7162717227 |

Задача Е. Муха

Ограничение по времени: 2 секунды
Ограничение по памяти: 64 Мб

Муха летит вдоль прямой. Если нанести на эту прямую координаты, то можно сказать, что в 0-й момент времени муха пролетает точку с координатой 0 и летит в положительном направлении со скоростью V . Муха может менять свою скорость, однако ускорение мухи не может по модулю превышать величины A , в частности, муха не может мгновенно остановиться. Максимальная скорость мухи не может превышать по модулю величины W .

Известно, что в момент времени T по прямой ударит мухобойка, которая полностью накроет отрезок от точки C до точки D . Если муха в этот момент окажется на этом отрезке, она погибнет.

Напишите программу, которая определит, есть ли у мухи шанс спастись, и если есть, то выведет, что должна муха для этого делать.

Формат входных данных

Во входном файле заданы числа V, W, A, T, C, D . Все числа целые. $0 \leq V \leq W \leq 1000$, $0 \leq A \leq 1000$, $0 < T \leq 1000$, $-1000000 \leq C \leq D \leq 1000000$.

Формат выходных данных

Если муха может спастись, выведите, как она должна для этого лететь. Для этого выведите последовательность команд для мухи. Количество команд не должно превышать 100. Каждая команда задается двумя числами T_i, A_i , которые обозначают, что в течение времени T_i муха должна лететь с ускорением A_i . T_i и A_i не обязаны быть целыми, T_i должны быть положительны (не могут быть равны 0), сумма всех T_i должна быть равна T с точностью до 10^{-6} .

Если, в рамках указанных ограничений, муха спастись не сможет, в выходной файл выведите одно число -1.

Примеры

| fly.in | fly.out |
|-----------------------|----------------|
| 10 10 5 1 -100 100 | -1 |
| 10 20 5 1 9 11 | 0.2 0 0.8 4 |
| 10 10 5 5 0 1000 | 4 -5 1 0 |