

## Задача А. Река

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

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

## Задача В. Пусти козла в огород - 8

Имя входного файла: `goat8.in`  
Имя выходного файла: `goat8.out`  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 64 мегабайта

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<br>11 4 2       | 0   |
| 3 4 5<br>9 4 2        | 2<br>7.7500000000 2.4387505004<br>7.7500000000 5.5612494996 |

## Задача С. Правильный многоугольник

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

Даны  $n$  пар вещественных чисел. Есть гипотеза, что это координаты вершин невырожденного правильного  $n$ -угольника, записанные в порядке обхода (либо в положительном, либо в отрицательном направлении). Проверьте, так ли это.

### Формат входных данных

В первой строке записано целое число  $n$  ( $3 \leq n \leq 100$ ). В  $i$ -й из следующих  $n$  строк через пробел записаны вещественные числа  $x_i$  и  $y_i$  ( $0 \leq x_i, y_i \leq 1$ ) — координаты  $i$ -й точки. Координаты различных точек могут совпадать, но гарантируется, что существует хотя бы одна пара точек на расстоянии не меньше 0.3. Координаты заданы с точностью не менее  $10^{-10}$ .

### Формат выходных данных

Если в результате эксперимента не удалось построить вершины правильного  $n$ -угольника в порядке обхода, выведите в единственной строке `NO`, в противном случае выведите `YES`. Гарантируется, что в случае отрицательного ответа нельзя изменить координаты точек менее чем на  $10^{-5}$  так, чтобы они стали координатами вершин правильного  $n$ -угольника, записанными в порядке обхода.

### Примеры

| <code>ngon.in</code>          | <code>ngon.out</code> |
|-------------------------------|-----------------------|
| 4<br>0 0<br>1 0<br>1 1<br>0 1 | YES                   |
| 3<br>0 0<br>1 0<br>0.5 1      | NO                    |

## Задача D. Точка в многоугольнике

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

### Формат входных данных

В первой строке содержится три числа —  $N$  ( $3 \leq N \leq 100000$ ) 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<br>1 1<br>10 2<br>2 8 | YES                    |

## Задача Е. Стена

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

Жил-был жадный король, который однажды приказал главному архитектору окружить королевский замок стеной. Король был настолько жаден, что не желал слушать рассказы архитектора о красивой кирпичной стене с прекрасным силуэтом и изящными высокими башнями. Вместо этого он приказал окружить замок стеной, затратив минимальное количество камня и времени, но потребовал, чтобы стена не подходила к замку ближе, чем на заданное расстояние. Если король узнает, что архитектор потратил не минимально возможное количество ресурсов, то архитектор лишится головы. Более того, король потребовал, чтобы архитектор сразу же предложил проект стены с указанием минимального количества ресурсов, необходимых для постройки.

Вы должны помочь архитектору сохранить голову, написав программу для поиска минимальной длины стены, удовлетворяющей условиям короля.

Задачу упрощает то, что замок короля имеет форму многоугольника и расположен на равнине. Архитектор уже ввел систему координат и точно измерил координаты вершин замка в футах.

### Формат входных данных

Первая строка входного файла содержит числа  $N$  и  $L$ , разделенные пробелом.  $N$  ( $3 \leq N \leq 1000$ ) — это количество вершин в королевском замке, а  $L$  ( $1 \leq L \leq 1000$ ) — минимальное количество футов, на которое стена может приближаться к замку.

Следующие  $N$  строк описывают координаты замка в порядке обхода по часовой стрелке. В каждой строке через пробел записаны целые числа  $x_i$  и  $y_i$ , разделенные пробелом ( $-10000 \leq x_i, y_i \leq 10000$ ), которые обозначают координаты  $i$ -ой вершины. Все вершины различны, и никакие две стороны не пересекаются кроме как по вершинам.

### Формат выходных данных

Выведите минимальную длину стены в футах, удовлетворяющей условиям короля с точностью не менее 6 знаков после запятой.

### Примеры

| <code>wall.in</code>   | <code>wall.out</code> |
|--|-----------------------|
| 9 100<br>200 400<br>300 400<br>300 300<br>400 300<br>400 400<br>500 400<br>500 200<br>350 200<br>200 200 | 1628.3185307180       |

## Задача F. Пусти оленя в огород

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

Мы уже научились справляться с козлами и огородами. Теперь, давайте, перенесемся немного севернее и разберемся с особенностями управления оленями. Оленевод Ванхо привязал своего оленя Ахтамака к изгороди нефтяной площадки, имеющей форму выпуклого многоугольника. Олень был привязан на длинной верёвке, чтобы он не убежал и при этом мог пастись. Вокруг нефтяной вышки растёт такой вкусный ягель, что олень тут же принялся его щипать. Напишите программу, вычисляющую площадь участка вне изгороди, ягель на которой будет доступен оленю. Форма изгороди, точка привязывания и длина верёвки задаются во входном файле.

### Формат входных данных

В первой строке входного файла записано целое число  $n$  — количество углов изгороди ( $3 \leq n \leq 100$ ). В последующих  $n$  строках записаны координаты углов изгороди в порядке обхода по часовой стрелке. В последней строке записаны три числа — координаты точки привязывания оленя к изгороди и длина верёвки. Все координаты целые и не превосходят по модулю  $10^4$ . Длина верёвки — целое положительное число, не превосходящее  $10^4$ . Числа в каждой строке разделены пробелами. Гарантируется, что изгородь представляет собой строго выпуклый многоугольник и точка привязывания оленя лежит на его границе. Также длина веревки не превосходит половины периметра изгороди.

### Формат выходных данных

В выходной файл выведите значение площади с точностью не менее  $10^{-3}$

### Примеры

| <code>deer.in</code>                       | <code>deer.out</code> |
|--|-----------------------|
| 4<br>-5 -5<br>-5 5<br>5 5<br>5 -5<br>5 0 4 | 25.1327412287         |