

## Задача А. Уравнение прямой I

Имя входного файла: line1.in  
Имя выходного файла: line1.out  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 64 мегабайта

### Формат входных данных

Четыре числа — целые координаты двух различных точек на прямой. ( $|x|, |y| \leq 10^4$ )

### Формат выходных данных

Три числа — дробные коэффициенты  $A$ ,  $B$  и  $C$  уравнения этой прямой с точностью до шестого знака после запятой. Выводите результат с точностью не менее 6 знаков после запятой.

### Примеры

line1.in	line1.out
1 2 3 1	-1 -2 5

## Задача В. Расстояние от точки до прямой

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

Найдите расстояние от заданной точки до заданной прямой.

### Формат входных данных

Пять целых чисел — координаты точки и коэффициенты  $A$ ,  $B$  и  $C$  нормального уравнения прямой.

### Формат выходных данных

Одно число — расстояние от точки до прямой с точностью не менее  $10^{-6}$ .

### Примеры

<code>distance1.in</code>	<code>distance1.out</code>
1 1 1 1 -1	0.7071067812

## Задача С. Пусти козла в огород -7

Имя входного файла: `goat7.in`  
Имя выходного файла: `goat7.out`  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

Петр Васильевич всегда выгуливает своих козлов у себя в огороде, но ему не нравится, что они вытаптывают его грядки, поэтому он решил усовершенствовать привязь. Теперь ошейник у них прикреплен к веревке, натянутой между двумя деревьями, поэтому козлы могут ходить только по отрезку, их соединяющему.

Но козлам скучно ходить по огороду по одиночке, поэтому они хотят встретиться и пожевать траву вместе. Помогите им встретиться!

### Формат входных данных

Программа получает на вход восемь целых чисел, по абсолютной величине не превосходящих  $10^4$ . Сначала указываются координаты двух деревьев, к которым привязан первый козел, а затем - координаты деревьев, к которым привязан второй. Заметим, что для привязи могут служить одни и те же деревья (в том числе могут совпадать и концы одной привязи).

### Формат выходных данных

Если козлам встретиться не суждено, выведите строку `Empty`. Если они могут встретиться только в одной точке, то выведите два числа — координаты точки пересечения. Если пересечением является отрезок, то выведите четыре числа — координаты двух концов отрезка в лексикографическом порядке (то есть сначала нужно вывести ту точку, у которой меньше координата  $x$ , а если у них равны координаты  $x$  то ту, у которой меньше координата  $y$ ). Все числа следует выводить с точностью не менее 6 знаков после запятой.

### Примеры

<code>goat7.in</code>	<code>goat7.out</code>
0 0 9 9 9 5 0 5	5.0000000000 5.0000000000
0 0 9 9 15 15 7 7	7.0000000000 7.0000000000 9.0000000000 9.0000000000
0 0 9 9 10 10 10 10	Empty

## Задача D. 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 Е. Две окружности

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

Вам заданы две окружности. Необходимо выяснить, пересекаются ли заданные окружности и найти точки их пересечения.

### Формат входных данных

Первая строка файлов содержит количество тестов — целое число от 1 до 10 000. Далее следуют тесты. Каждый тест имеет вид: две строки, по три числа в каждой — координаты центра и радиус сначала первой, а затем второй окружности.

Все числа — целые, по модулю не превосходящие 10 000. А радиус еще и положительный.

### Формат выходных данных

На каждый тест выведите ответ в следующем формате:

На отдельной строке выходного файла выведите количество точек пересечения (0, 1, 2 или 3, если их бесконечно много). В случае одной точки пересечения выведете во второй строке координаты этой точки. В случае двух точек пересечения выведете во второй строке координаты точки  $H$ , в третьей строке длины векторов  $\overrightarrow{O_1H}$  и  $\overrightarrow{HP}$ , в следующих двух строках должны находиться координаты точек пересечения. Эти две точки можно вывести в произвольном порядке

### Примеры

<code>intersec.in</code>	<code>intersec.out</code>
4	0
3 4 5	1
11 4 2	8.0000000000 4.0000000000
3 4 5	2
11 4 3	7.5625000000 4.0000000000
3 4 5	4.5625000000 2.0453835215
11 4 4	7.5625000000 6.0453835215
3 4 5	7.5625000000 1.9546164785
3 4 5	3

## Задача F. 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  $P_1$ . 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 выведите во второй строке координаты точки  $P_3$ , в третьей длины векторов  $\overrightarrow{P_1P_3}$  и  $\overrightarrow{P_3P_2}$ , в следующих 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

## Задача G. Биссектриса

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

Найдите прямую, содержащую биссектрису угла, заданного вершиной  $X$  и двумя точками  $Y$  и  $Z$  на его сторонах.

### Формат входных данных

Шесть целых чисел, не превышающих по модулю  $10^4$  — координаты точек  $X$ ,  $Y$  и  $Z$ .

### Формат выходных данных

Три числа — коэффициенты нормального уравнения биссектрисы угла  $\angle YXZ$  с точностью до шести знаков после запятой.

### Примеры

<code>bisector.in</code>	<code>bisector.out</code>
1 1 1 0 0 1	-1.0 1.0 -0.0