

## Задача А. Угол между векторами

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

Необходимо найти угол между двумя векторами на плоскости в радианах.

### Формат входных данных

В первой строке задано четыре целых числа — координаты двух невырожденных векторов, не превосходящие 10 000 по абсолютной величине.

### Формат выходных данных

Выведите одно число — величину неориентированного угла между ними с точностью до пятого знака после десятичной точки из отрезка  $[0, \pi]$ .

### Пример

angle.in	angle.out
2 1 3 5	0.56673

## Задача В. Площадь выпуклого многоугольника

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

Многоугольник называется *выпуклым*, если любой отрезок, соединяющий две точки внутри или на сторонах многоугольника, не выходит за границы многоугольника.

Например, любой треугольник является выпуклым многоугольником, а «звезда» не является.

Дан выпуклый многоугольник. Найдите его площадь.

### Формат входных данных

В первой строке находится одно целое число  $N$  — количество вершин многоугольника. Далее следуют  $N$  строк, каждая из которых содержит два целых числа  $x_i, y_i$  — координаты вершины многоугольника.

Вершины перечислены последовательно, в порядке обхода по часовой стрелке или против неё. Ограничения:  $|x_i|, |y_i| \leq 10\,000, 3 \leq N \leq 100$ .

### Формат выходных данных

В первой строке выведите вещественное число — площадь многоугольника. Абсолютная погрешность не должна превышать  $10^{-6}$ .

## Примеры

area.in	area.out
3 0 0 0 1 2 0	1
4 0 0 0 1 1 1 1 0	1

## Задача С. Центр

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

*Расстоянием Чебышёва*  $\rho_\infty(A, B)$  между двумя точками  $A = (x_1, y_1)$  и  $B = (x_2, y_2)$  на плоскости называется максимум из модулей разностей координат этих точек:

$$\rho_\infty(A, B) = \max\{|x_2 - x_1|, |y_2 - y_1|\}.$$

Символ  $\rho_\infty$  используется потому, что расстояние Чебышёва является предельным для целого класса расстояний:

$$\rho_k(A, B) = (|x_2 - x_1|^k + |y_2 - y_1|^k)^{\frac{1}{k}}.$$

Так,  $\rho_1(A, B) = |x_2 - x_1| + |y_2 - y_1|$  — манхэттенское расстояние,  $\rho_2(A, B) = \sqrt{|x_2 - x_1|^2 + |y_2 - y_1|^2}$  — обычное евклидово расстояние.

Расстояние Чебышёва называют ещё *шахматным расстоянием*: на шахматной доске это расстояние — количество ходов короля между двумя клетками.

Назовём *центром* множества точек на плоскости такую точку, сумма расстояний Чебышёва от которой до всех точек множества минимальна. Найдите центр заданного множества точек.

### Формат входных данных

В первой строке ввода задано целое число  $n$  — количество точек

( $1 \leq n \leq 50$ ). Следующие  $n$  строк содержат координаты точек; в  $i$ -й из них записаны два целых числа  $x_i$  и  $y_i$  через пробел ( $|x_i|, |y_i| \leq 10\,000$ ).

### Формат выходных данных

В первой строке выведите одно число — минимальное суммарное расстояние Чебышёва от всех данных точек до найденного центра. Во второй строке выведите координаты центра. Выводите числа не менее чем с шестью точными знаками после десятичной точки. Если возможных положений центра несколько, можно вывести любое из них.

### Примеры

center.in	center.out
1 1 1	0 1 1
2 1 1 2 2	1 1.5 1.5
3 1 3 2 1 3 2	2.500000 2.500000 1.500000

## Задача D. Две окружности

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

Заданы две окружности на плоскости. Задача заключается в нахождении всех точек их пересечения.

### Формат входных данных

Первая строка содержит число наборов входных данных  $K$  ( $1 \leq K \leq 10\,000$ ). Каждый набор состоит из двух строчек, каждая из которых описывает окружность. Описание окружности задаётся в виде трёх чисел  $x$ ,  $y$  и  $r$ , разделённых пробелами ( $-100 \leq x, y \leq 100$ ,  $0 < r \leq 100$ ). Все числа во вводе целые.

### Формат выходных данных

Для каждого из наборов необходимо вывести одно из нижеследующих сообщений:

- «There are no points!!!» — если точки пересечения отсутствуют.
- «There are only  $i$  of them....» — если окружности имеют в точности  $i$  точек пересечения. В этом случае последующие  $i$  строчек должны содержать координаты точек  $x'_j$  и  $y'_j$ . Точки требуется выводить в порядке возрастания (сначала с меньшими  $x$ , если значения  $x$  равны, то с меньшими  $y$ ). Числа необходимо выводить не менее чем с четырьмя точными знаками после десятичной точки.
- «I can't count them - too many points :(» — если точек пересечения бесконечно много.

Все сообщения необходимо выводить без кавычек.  
 Разделяйте вывод для различных наборов пустой строкой.

### Пример

circles.in
2 0 0 2 4 0 2 0 0 1 100 100 1
circles.out
There are only 1 of them.... 2.0000000000000000 0.0000000000000000
There are no points!!!

## Задача E. Классификация четырёхугольников

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

Заданы четыре точки  $A$ ,  $B$ ,  $C$  и  $D$  на плоскости. Известно, что никакие три из этих точек не лежат на одной прямой. Проведём четыре отрезка  $AB$ ,  $BC$ ,  $CD$  и  $DA$ , получив замкнутую ломаную  $ABCD$ . Какой четырёхугольник задаёт эта ломаная?

Ниже перечислены классы четырёхугольников, которые следует различать в этой задаче.

- Если  $ABCD$  — это квадрат (все стороны равны, все углы прямые), ответом будет слово «square».
- Если  $ABCD$  — это ромб (все стороны равны), но не квадрат, ответом будет слово «rhombus».
- Если  $ABCD$  — это прямоугольник (все углы прямые), но не квадрат, ответом будет слово «rectangle».
- Если  $ABCD$  — это параллелограмм (противоположные стороны параллельны), но не прямоугольник и не ромб, ответом будет слово «parallelogram».
- Если  $ABCD$  — это трапеция (выпуклый четырёхугольник, не имеющий самопересечений, две стороны которого параллельны), но не параллелограмм, ответом будет слово «trapezoid».
- Если  $ABCD$  — это выпуклый четырёхугольник, не имеющий самопересечений (все внутренние углы меньше развёрнутого угла), но не трапеция, ответом будет словосочетание «convex polygon».
- Если  $ABCD$  — это невыпуклый четырёхугольник, не имеющий самопересечений (какой-то из внутренних углов больше развёрнутого угла), ответом будет словосочетание «non-convex polygon».
- Наконец, если  $ABCD$  — это самопересекающаяся ломаная (какие-то два проведённых отрезка пересекаются), ответом будет словосочетание «self-intersecting polyline».

### Формат входных данных

В каждой из четырёх строк ввода задано по два числа  $x$  и  $y$  — координаты очередной вершины ломаной ( $-100 \leq x, y \leq 100$ ). В первой строке заданы координаты точки  $A$ , во второй — точки  $B$ , в третьей — точки  $C$ , а в четвёртой — точки  $D$ . Гарантируется, что никакие три заданные точки не лежат на одной прямой.

### Формат выходных данных

Выведите слово или словосочетание, являющееся ответом.

### Примеры

<b>classify.in</b>	<b>classify.out</b>
0 0 0 1 1 1 1 0	square
2 1 -2 -1 -3 1 1 3	rectangle

### Пояснения к примерам

В первом примере  $ABCD$  — это квадрат со стороной 1.

Во втором примере  $ABCD$  — это прямоугольник со сторонами  $\sqrt{5}$  и  $2 \cdot \sqrt{5}$ .

### Задача F. Выпуклая оболочка

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

Выпуклая оболочка множества точек на плоскости — это выпуклый многоугольник минимальной площади такой, что все точки множества лежат у него внутри или на границе. Известно, что для любого множества точек выпуклая оболочка единственна с точностью до порядка обхода её вершин.

По данному множеству из  $n$  точек постройте его выпуклую оболочку.

### Формат входных данных

В первой строке задано число  $n$  ( $3 \leq n \leq 100$ ). Следующие  $n$  строк содержат по два числа  $x_i$   $y_i$  каждая через пробел — координаты  $i$ -й точки ( $-1000 \leq x_i, y_i \leq 1000$ ). Все заданные числа целые.

### Формат выходных данных

В первой строке выведите  $k$  — количество вершин выпуклой оболочки. В следующих  $k$  строках выведите по два числа  $x'_k$   $y'_k$  через пробел — координаты вершин многоугольника, являющегося выпуклой оболочкой, в порядке их обхода по часовой стрелке. Каждый из углов многоугольника должен быть строго больше нуля и строго меньше 180 градусов.

Гарантируется, что заданные точки таковы, что такой многоугольник существует.

### Примеры

<code>convex.in</code>	<code>convex.out</code>
3 0 0 1 1 0 1	3 0 0 0 1 1 1
6 0 0 0 2 2 0 2 2 2 1 1 1	4 0 2 2 2 2 0 0 0

### Задача G. Расстояние

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

Дан невырожденный треугольник. Найдите расстояние между точкой пересечения его высот (или их продолжений) и точкой пересечения серединных перпендикуляров.

#### Формат входных данных

В первой строке находятся шесть целых чисел — координаты трёх вершин треугольника.

Гарантируется, что три вершины не лежат на одной прямой и удовлетворяют ограничениям:  $|x_i|, |y_i| \leq 10\,000$ .

#### Формат выходных данных

В первой строке выведите единственное вещественное число — расстояние между точкой пересечения его высот (или их продолжений) и точкой пересечения серединных перпендикуляров. Абсолютная погрешность не должна превышать  $10^{-6}$ .

### Примеры

<code>distance.in</code>	<code>distance.out</code>
0 0 120 160 240 0	55
0 0 0 2 3 1	1

### Задача H. Точка в многоугольнике

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

В этой задаче необходимо выяснить, верно ли, что заданная точка находится внутри или на границе заданного многоугольника.

#### Формат входных данных

В первой строке задано три числа —  $N$  ( $3 \leq N \leq 100\,000$ ) и координаты точки. Далее  $N$  строк содержат по два числа каждая — координаты очередной вершины простого многоугольника в порядке обхода по или против часовой стрелки. Все координаты целые и не превосходят 10 000 по абсолютной величине.

#### Формат выходных данных

Выведите «YES», если заданная точка содержится в приведённом многоугольнике или на его границе, и «NO» в противном случае.

### Пример

<code>inside.in</code>	<code>inside.out</code>
3 0 0 1 0 0 1 1 1	NO

### Задача I. Воздушные змеи

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

Назовём *воздушным змеем* четырёхугольник, состоящий из двух равных прямоугольных треугольников, склеенных по гипотенузе так, чтобы получилась симметричная фигура относительно линии склейки. Эту линию будем называть *осью змея*.

На плоском столе лежало несколько воздушных змеев из бумаги. Каждый из них разрезали по оси. После этого получившиеся треугольники, возможно, подвинули и повернули (но не отразили) независимо друг от друга. Наконец, некоторые треугольники убрали со стола.

Зная, где находятся оставшиеся треугольники, выясните, какое минимальное количество воздушных змеев могло быть на столе исходно, а также из каких треугольников они могли состоять. Треугольники на столе могут накладываться произвольно.

### Формат входных данных

В первой строке записано целое число  $n$  — количество треугольников ( $1 \leq n \leq 100\,000$ ). Каждая из следующих  $n$  строк содержит шесть чисел  $x_1, y_1, x_2, y_2, x_3, y_3$  — координаты трёх вершин очередного треугольника. Вершины треугольника могут быть перечислены в произвольном порядке. Гарантируется, что все треугольники — прямоугольные, а все координаты — целые числа, не превосходящие  $10^6$  по абсолютной величине.

### Формат выходных данных

В первой строке выведите число  $m$  — минимальное количество воздушных змеев, которые могли находиться на столе исходно. Далее выведите возможное описание  $m$  воздушных змеев в любом порядке, по одному змею в строке. Каждое описание должно иметь вид  $k_1-k_2$ . Здесь  $k_1$  и  $k_2$  — номера треугольников, из которых состоит воздушный змей, в любом порядке. Треугольники на столе пронумерованы целыми числами от 1 до  $n$  в порядке их следования во входных данных. Убранные со стола треугольники обозначаются номером 0. Каждый треугольник на столе должен встречаться в описании ровно один раз.

Если возможных ответов с минимальным  $m$  несколько, выведите любой из них.

### Пример

kites.in	kites.out
6	4
0 0 0 1 1 1	3-4
0 0 1 0 1 1	0-5
1 1 1 2 3 2	2-1
0 0 0 2 1 2	6-0
3 4 0 0 -8 6	
0 5 0 0 -10 0	

## Задача J. Точка пересечения прямых

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

Даны две прямые на плоскости. Каждая прямая задаётся двумя точками. Найдите точку пересечения данных прямых.

### Формат входных данных

Ввод состоит из двух строк. Каждая из них задаёт одну прямую и содержит четыре целых числа  $x_1, y_1, x_2, y_2$  — координаты двух различных точек, через которые проходит прямая ( $|x_1|, |y_1|, |x_2|, |y_2| \leq 10\,000$ ).

### Формат выходных данных

Если точка пересечения прямых существует и единственна, выведите в первой строке два вещественных числа — координаты точки пересечения. Абсолютная погрешность не должна превышать  $10^{-6}$ .

Если точки пересечения не существует, выведите в первой строке единственное число: 0.

Если прямые совпадают, выведите  $-1$ .

### Примеры

lines.in	lines.out
0 0 1 1 0 1 1 0	0.5 0.5
1 5 5 5 2 5 1 5	-1

## Задача K. Пары монстров

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

На острове Дженту обитают монстры. Каждый монстр характеризуется своей уродливостью и своей отвратительностью. Уродливость монстра — целое число, не превосходящее по модулю  $10^9$ ; если это число отрицательно, значит, монстр красив. Отвратительность монстра — также целое число, не превосходящее по модулю  $10^9$ ; если это число отрицательно, значит, монстр привлекателен.

Два монстра нравятся друг другу, если произведение их уродливостей — величина, противоположная произведению их отвратительностей. Формально, если один монстр имеет уродливость  $a_1$  и отвратительность  $b_1$ , а другой монстр — уродливость  $a_2$  и отвратительность  $b_2$ , эти монстры нравятся друг другу тогда и только тогда, когда  $a_1 \times a_2 = -(b_1 \times b_2)$ .

По заданным характеристикам монстров найдите количество различных пар монстров, которые нравятся друг другу. Пары, в которых первый и второй монстры совпадают, также следует учитывать. Пара  $(i, j)$  и пара  $(j, i)$  считаются одинаковыми.

### Формат входных данных

В первой строке ввода содержится одно целое число  $n$  — количество монстров на острове ( $1 \leq n \leq 200\,000$ ). Следующие  $n$  строк описывают монстров. Каждая строка содержит два целых числа  $a_i$  и  $b_i$ , разделённых пробелом — уродливость и отвратительность  $i$ -го монстра ( $|a_i|, |b_i| \leq 10^9$ ).

### Формат выходных данных

Выведите одно число — количество различных пар монстров, которые нравятся друг другу.

### Примеры

monster-pairs.in	monster-pairs.out
4 -1 -1 -2 3 6 4 12 -18	2
2 0 0 0 0	3

### Пояснения к примерам

В первом примере второй и третий монстры нравятся друг другу, так как  $-2 \times 6 = -(3 \times 4)$ . Кроме того, третий и четвёртый монстры также нравятся друг другу, поскольку  $6 \times 12 = -(-4 \times -18)$ . Других пар монстров, которые нравятся друг другу, нет. Две искомые пары монстров —  $(2, 3)$  и  $(3, 4)$ .

Во втором примере уродливость и отвратительность обоих монстров равны нулю. Согласно формальному определению, каждый из них нравится и сам себе, и другому монстру. Три искомые пары монстров —  $(1, 1)$ ,  $(1, 2)$  и  $(2, 2)$ .

## Задача L. Принадлежность точки отрезку

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

Требуется проверить, принадлежит ли заданная точка заданному отрезку (концы включаются в отрезок) на плоскости.

### Формат входных данных

В первой строке задано есть целых чисел — координаты точки и координаты концов отрезка, не превосходящие 10 000 по абсолютной величине.

### Формат выходных данных

Выведите «YES», если точка принадлежит отрезку, и «NO» в противном случае.

### Пример

point.in	point.out
3 3 1 2 5 4	YES

## Задача M. Космическая полиция

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

Полицейский катер, патрулирующий звёздную систему, обнаружил корабль пиратов. Переговоры ни к чему не привели, и капитан Синтия решила пойти на abordаж.

Карту космоса вокруг двух кораблей можно упрощённо представить в виде плоскости, а корабли в виде точек на этой плоскости. Координаты и расстояния на карте измеряются в космических милях.

Корабль пиратов находится в точке плоскости  $(x_1, y_1)$  и движется по прямой со скоростью одной космической мили в минуту. Известно, что через минуту корабль пиратов должен оказаться в точке  $(x_2, y_2)$ .

Полицейский катер находится в точке плоскости  $(x_3, y_3)$  и может двигаться в любую сторону с той же скоростью, что и корабль пиратов — одна космическая миля в минуту. Синтия хочет проложить курс катера так, чтобы он встретился с кораблём пиратов как можно скорее. Чтобы спланировать abordаж, капитан хочет знать, через какое время в этом случае произойдёт

встреча. Помогите ей узнать это время или определите, что встретиться с кораблём пиратов не удастся.

### Формат входных данных

В первой строке заданы два целых числа  $x_1$  и  $y_1$ , разделённые пробелом — координаты корабля пиратов.

Во второй строке заданы два целых числа  $x_2$  и  $y_2$ , разделённые пробелом — ожидаемое положение корабля пиратов через одну минуту.

В третьей строке заданы два целых числа  $x_3$  и  $y_3$ , разделённые пробелом — координаты полицейского катера.

Все координаты — целые числа, не превосходящие 100 по абсолютной величине.

Гарантируется, что точки  $(x_1, y_1)$  и  $(x_2, y_2)$  лежат на расстоянии ровно в одну космическую милю. Кроме того, гарантируется, что начальные положения корабля пиратов и полицейского катера не совпадают.

### Формат выходных данных

Выведите одно вещественное число — минимальное время в минутах, через которое полицейский катер сможет оказаться в одной и той же точке с кораблём пиратов. Если это невозможно, выведите число  $-1$ .

Пожалуйста, убедитесь, что найденный вами ответ выведен достаточно точно. Ваш ответ будет считаться верным, если он отличается от точного ответа не более чем на одну тысячную часть минуты.

### Примеры

<code>police.in</code>	<code>police.out</code>
1 1 2 1 4 4	3.0000000000
0 0 0 1 0 -2	-1

## Задача N. Распознай-ка

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

Тебе еще многому нужно учиться, юный падаван

Оби-Ван Кеноби

На плоскости были нарисованы  $N$  отрезков с целыми координатами концов. Длины отрезков были не меньше 50. Для любой пары отрезков угол между прямыми, образованными отрезками, был больше  $\frac{\pi}{8} - 10^{-6}$ . Из этого, например, следует, что  $N \leq 8$ .

Теперь есть координатная сетка, столбцы которой пронумерованы от 0 до  $W - 1$ , а строки — от 0 до  $H - 1$ . Клеткам сетки соответствуют точки на плоскости с такими же координатами. Изначально все клетки сетки белые.

$N$  отрезков нарисованы на сетке чёрным цветом. Клетка покрашена в чёрный цвет, если существует хотя бы один отрезок, расстояние от которого до клетки не превосходит 1.

Отрезки целиком поместились на сетку (если бы сетка содержала все целые точки плоскости от  $-\infty$  до  $\infty$ , никакие точки, кроме уже покрашенных, не пришлось бы красить).

### Формат входных данных

В первой строке ввода даны два числа  $H$  и  $W$  — высота и ширина сетки ( $1 \leq H \leq 300$ ,  $1 \leq W \leq 300$ ). Далее идут  $H$  строк по  $W$  символов — сама координатная сетка. Символ «\*» обозначает чёрную краску, а символ «.» — белую краску.

### Формат выходных данных

Выведите одно число — количество отрезков в исходном наборе.



### Примеры

segments.in	segments.out
0 0 9 9 9 5 0 5	5.000000 5.000000
0 0 9 9 15 15 7 7	7.000000 7.000000 9.000000 9.000000
0 0 9 9 10 10 10 10	Empty

### Задача Р. Лыжная трасса

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

Лыжница Аня находится на старте лыжной трассы. Трасса представляет собой участок плоскости, на котором введена декартова система координат. Координаты измеряются в метрах. Старт соответствует прямой  $y = 0$ , а финиш — прямой  $y = n$ . На прямых  $y = 1, y = 2, \dots, y = n$  расположены ворота. Каждые ворота состоят из двух флажков, расстояние между которыми равно  $d$  метрам. Ворота характеризуются координатой левого края: на каждой прямой  $y = k$  эта координата равна  $x_k$ .

Чтобы пройти трассу, Ане нужно последовательно пройти сквозь все ворота. Другими словами, пересекать каждую прямую вида  $y = k$  можно лишь в точках  $(x, y)$  таких, что  $x_k \leq x \leq x_k + d$ . Движение между двумя соседними воротами ничем не ограничено.

Аня может начать движение из любой точки прямой  $y = 0$  и закончить движение в любой точке прямой  $y = n$ , находящейся внутри последних ворот. Найдите минимальную длину маршрута, по которому можно пройти трассу. Размером Ани можно пренебречь и считать её точкой.

#### Формат входных данных

В первой строке заданы через пробел два целых числа  $n$  и  $d$  — количество

ворот и ширина ворот в метрах ( $1 \leq n \leq 100, 2 \leq d \leq 10$ ). В следующей строке заданы  $n$  целых чисел  $x_1, x_2, \dots, x_n$  через пробел — координаты левого края первых, вторых, ..., последних ворот трассы ( $0 \leq x_k \leq 1000$ ).

#### Формат выходных данных

Выведите одно число — минимальную длину пути по трассе с точностью не менее шести знаков после десятичной точки.

### Примеры

ski.in	ski.out
3 2 1 2 1	3
5 2 0 2 5 7 10	9.94427190999916

### Задача Q. Квадрат

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

На уроке геометрии при обсуждении задачи о квадратуре круга семи-классник Иван заявил, что может сделать квадрат из чего угодно. Владимир Алексеевич, учитель геометрии, тут же решил проверить это. Он нарисовал на доске прямоугольную систему координат и отметил две различных точки:  $A$  и  $B$ . Задача Ивана — отметить ещё две точки  $C$  и  $D$  так, чтобы четыре отмеченные точки лежали в вершинах какого-то квадрата.

Помогите Ивану выбрать такие точки.

#### Формат входных данных

Первая строка входных данных содержит два числа  $x_A$  и  $y_A$  — координаты точки  $A$ . Вторая строка содержит два числа  $x_B$  и  $y_B$  — координаты точки  $B$ . Все заданные числа целые и лежат в пределах от 0 до 100. Гарантируется, что заданные точки различны.

#### Формат выходных данных

Выведите в первой строке два вещественных числа  $x_C$  и  $y_C$  — координаты точки  $C$ . Во второй строке выведите два вещественных числа  $x_D$  и  $y_D$  — координаты точки  $D$ . Числа в строке разделяйте пробелом. Точки  $A, B, C$  и  $D$  должны лежать в вершинах некоторого квадрата. Если возможных ответов несколько, выведите любой из них.

### Пример

square.in	square.out
1 0	1 1
0 1	0 0

### Задача R. Площадь треугольника

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

Даны три точки на плоскости. Какова площадь треугольника с вершинами в этих точках?

### Формат входных данных

В первых трёх строках входных данных заданы координаты точек. Каждая из этих строк содержит два целых числа  $x_i$  и  $y_i$ , разделённых пробелом ( $|x_i|, |y_i| \leq 100$ ). Учтите, что вершины треугольника могут лежать на одной прямой и даже совпадать.

### Формат выходных данных

В первой строке выведите одно число — площадь треугольника с вершинами в данных точках. Выводите ответ с точностью не менее шести знаков после десятичной точки.

### Примеры

tarea.in	tarea.out
0 0 0 1 1 1	0.5
-2 -2 0 0 2 3	1

### Задача S. Треугольник и точка

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

Даны треугольник и точка на плоскости. Определите, лежит ли точка

внутри, на границе или вне треугольника.

### Формат входных данных

В первых трёх строках заданы координаты вершин треугольника. Каждая из этих строк содержит два целых числа  $x_i$  и  $y_i$ , разделённых пробелом. В четвёртой строке заданы координаты точки в аналогичном формате. Все координаты не превосходят по модулю 100. Учтите, что вершины треугольника могут лежать на одной прямой и даже совпадать.

### Формат выходных данных

Если точка лежит на стороне или в вершине треугольника, выведите «BORDER». Если точка лежит строго внутри треугольника, выведите «INSIDE». Наконец, если точка лежит вне треугольника, выведите «OUTSIDE».

### Примеры

tinside.in	tinside.out
0 0 0 2 1 1 0 1	BORDER
0 0 0 2 2 1 1 1	INSIDE
0 0 0 1 1 1 3 2	OUTSIDE

### Задача T. Два прямоугольника

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

В этой задаче нужно найти два прямоугольника заданной суммарной площади, имеющих минимальный суммарный периметр.

Как известно, площадь прямоугольника с длинами сторон  $m$  и  $n$  равна  $m \cdot n$ , а его периметр равен  $2 \cdot (m + n)$ .

Задано целое число  $s \geq 2$ . Рассмотрим два прямоугольника с целыми положительными длинами сторон, сумма площадей которых равна  $s$ . Какой может быть минимальная сумма периметров этих двух прямоугольников?

Формально следует выбрать четыре целых положительных длины сторон  $a$ ,  $b$ ,  $c$  и  $d$  так, чтобы суммарная площадь  $a \cdot b + c \cdot d$  была равна  $s$ , а суммарный периметр  $2 \cdot (a + b) + 2 \cdot (c + d)$  был минимально возможным.

### Формат входных данных

В единственной строке ввода задано одно число  $s$  ( $2 \leq s \leq 10^{18}$ ).

### Формат выходных данных

В первой строке выведите одно число: минимальный суммарный периметр. Во второй строке выведите через пробел два числа  $a$  и  $b$  — длины сторон первого прямоугольника. В третьей строке выведите через пробел два числа  $c$  и  $d$  — длины сторон второго прямоугольника. Если возможных ответов несколько, выведите любой из них.

### Примеры

two-rectangles.in	two-rectangles.out
5	12 1 1 2 2
8	16 3 2 1 2

### Пояснения к примерам

В первом примере единственный оптимальный ответ — квадраты размеров  $1 \times 1$  и  $2 \times 2$ . Их можно вывести в любом порядке.

Во втором примере существует и другой оптимальный ответ — вместо прямоугольников  $1 \times 2$  и  $2 \times 3$  можно выбрать два квадрата размера  $2 \times 2$  каждый.