

## Задача А. Векторы

Имя входного файла: `stdin`  
Имя выходного файла: `stdout`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайта

Даны два ненулевых вектора. Требуется вычислить:

- Длину первого и второго вектора (два числа)
- Вектор, образованный сложением данных двух векторов
- Первый вектор, умноженный на длину второго вектора
- Нормализованный второй вектор
- Площадь треугольника, построенного из этих векторов

### Формат входных данных

В двух строках входного файла заданы по четыре целых числа, не превосходящих по модулю 10 000, — координаты начала и конца первого вектора, затем второго.

### Формат выходных данных

В каждой строке выходного файла — ответ на соответствующий пункт задачи с точностью не менее  $10^{-6}$ .

### Примеры

<code>stdin</code>	<code>stdout</code>
5 1 2 6	5.8309518948453007425
1 1 7 8	9.2195444572928870741
	3 12
	-27.658633371878661222
	46.097722286464431818
	0.65079137345596849151
	0.75925660236529657343
	25.5

## Задача В. Принадлежность точки лучу

Имя входного файла: point2.in  
Имя выходного файла: point2.out  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 64 мегабайта

### Формат входных данных

Шесть чисел — координаты точки и координаты начала и конца вектора.

### Формат выходных данных

Одна строка «YES», если точка принадлежит лучу, определяемому вектором, и «NO» в противном случае.

### Примеры

point2.in	point2.out
1 6 3 7 5 8	NO

## Задача С. Положение точек вне прямой

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

Даны две точки и прямая. Определите, правда ли, что эти точки лежат по одну сторону от этой прямой. Гарантируется, что ни одна из точек не лежит на прямой.

### Формат входных данных

Семь чисел — координаты двух точек вне прямой и коэффициенты  $A$ ,  $B$  и  $C$  её нормального уравнения.

### Формат выходных данных

Одна строка «YES», если точки лежат по одну сторону прямой и «NO» в противном случае.

### Примеры

<code>position.in</code>	<code>position.out</code>
0 0 2 4 2 -1 -1	YES

## Задача D. Принадлежность точки отрезку

Имя входного файла: point3.in  
Имя выходного файла: point3.out  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 64 мегабайта

### Формат входных данных

Шесть целых чисел от  $-10^4$  до  $10^4$  — координаты точки и координаты концов отрезка.

### Формат выходных данных

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

### Примеры

point3.in	point3.out
3 3 1 2 5 4	YES

## Задача Е. Площадь многоугольника

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

На плоскости заданы координаты вершин многоугольника в порядке их обхода. Многоугольник не обязательно выпуклый, но не содержит самопересечений. Требуется найти его площадь.

### Формат входных данных

Сначала записано число  $N$  — количество вершин многоугольника ( $3 \leq N \leq 100$ ), затем  $N$  пар вещественных чисел, задающих координаты его вершин  $x_i, y_i$  ( $-1000 \leq x_i, y_i \leq 1000$ ).

### Формат выходных данных

Выведите площадь многоугольника не меньше, чем с 3 знаками после десятичной точки.

### Примеры

area.in	area.out
4 0 0 0 2 4 3.5 4 0	11.0

### Замечание

Если выводить вещественные числа как `print(x)`, то иногда они будут странно отформатированы, например,  $10^{-6} = \frac{1}{1000000}$  будет выведено как `1e-6`.

Поэтому числа с заданной точностью следует переводить в строку так:

```
x = 1.34
```

```
print("{:.6f}".format(x)) # строка "1.340000"
```

Для того, чтобы избежать проблем с погрешностью вашего ответа, если условие это позволяет, следует выводить числа с максимально возможной точностью, для типа `float` в Питоне это 16 десятичных знаков:

```
print("{:.16f}".format(x))
```

## Задача F. Принадлежность точки выпуклому многоугольнику

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

Задан выпуклый многоугольник и точка. Нужно определить, лежит ли точка внутри этого многоугольника.

### Формат входных данных

Задано число  $N$  ( $3 \leq N \leq 100$ ). Далее идет  $N$  пар вещественных чисел, задающих координаты вершин многоугольника. Последние два вещественных числа задают координаты точки. Гарантируется, что точка не лежит на границе многоугольника. Координаты имеют значения от  $-10^{12}$  до  $10^{12}$ .

### Формат выходных данных

В выходной файл выведите сообщение YES, если точка лежит внутри многоугольника или NO, вне него.

### Примеры

<code>point.in</code>	<code>point.out</code>
3 0 0 1 0 0 1.5 10 10	NO

## Задача G. Пересечение прямоугольников

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

На плоскости задано  $N$  прямоугольников с вершинами в точках с целыми координатами и сторонами, параллельными осям координат. Необходимо найти прямоугольник, являющийся их пересечением.

То, что это прямоугольник, докажите самостоятельно.

### Формат входных данных

В первой строке входного файла указано число  $N$  ( $1 \leq N \leq 1500$ ). В следующих  $N$  строках заданы по 4 целых числа  $x_1, y_1, x_2, y_2$  — сначала координаты левого нижнего угла прямоугольника, потом правого верхнего ( $-10^9 \leq x_1 \leq x_2 \leq 10^9, -10^9 \leq y_1 \leq y_2 \leq 10^9$ ). Обратите внимание, что прямоугольники могут вырождаться в отрезки и даже в точки.

### Формат выходных данных

В единственную строку выходного файла поместите описание искомого прямоугольника в том же формате, в котором заданы прямоугольники во входном файле.

Если пересечение заданных прямоугольников пусто, выведите в выходной файл единственное число -1.

### Примеры

<code>rect.in</code>	<code>rect.out</code>
2 0 0 2 2 1 1 3 3	1 1 2 2

## Задача Н. Две окружности

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

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

### Формат входных данных

Первая строка входного файла содержит число наборов входных данных  $K$  ( $1 \leq K \leq 10000$ ). Каждый набор состоит из двух строчек, каждая из которых описывает окружность. Описание окружности задается в виде трех чисел,  $x, y, r$  ( $-1000 \leq x, y \leq 1000, 0 < r \leq 1000$ ), разделенных пробелами. Все числа во входном файле 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.
- “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.

### Примеры

<code>circles.in</code>	<code>circles.out</code>
<code>2</code> <code>0 0 2</code> <code>4 0 2</code> <code>0 0 1</code> <code>1000 1000 1</code>	<code>There are only 1 of them....</code> <code>2.0 0.0</code>  <code>There are no points!!!</code>