

## Задача А. Угол между векторами

Имя входного файла: *стандартный ввод*  
Имя выходного файла: *стандартный вывод*  
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

<i>стандартный ввод</i>	<i>стандартный вывод</i>
2 1 3 5	0.56673

## Задача В. Точка пересечения прямых

Имя входного файла: *стандартный ввод*  
Имя выходного файла: *стандартный вывод*  
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

<i>стандартный ввод</i>	<i>стандартный вывод</i>
0 0 1 1 0 1 1 0	0.5 0.5
1 5 5 5 2 5 1 5	-1

## Задача С. Пересечение отрезков

Имя входного файла: *стандартный ввод*  
Имя выходного файла: *стандартный вывод*  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Два отрезка  $[A, B]$  и  $[C, D]$  на плоскости заданы координатами своих концов — точек  $A, B, C, D$ , имеющих координаты  $(X_a, Y_a), (X_b, Y_b), (X_c, Y_c)$  и  $(X_d, Y_d)$ .

Требуется найти пересечение этих отрезков и вывести:

- слово `Empty`, если эти отрезки не пересекаются;
- координаты точки пересечения, если пересечение состоит из единственной точки;
- координаты точек — начала и конца отрезка пересечения в лексикографическом порядке, если пересечение заданных отрезков — отрезок. То есть первой следует выводить точку с минимальным  $x$ , а если координаты  $x$  равны, то с минимальным  $y$ .

### Формат входных данных

Во входных данных содержатся восемь целых чисел, не превосходящих 1000 по абсолютной величине — координаты  $(X_a, Y_a), (X_b, Y_b), (X_c, Y_c)$  и  $(X_d, Y_d)$ . Отрезки могут быть вырожденными.

### Формат выходных данных

Числовые значения в ответе должны отличаться от правильных не более чем на  $10^{-6}$ .

### Примеры

<i>стандартный ввод</i>	<i>стандартный вывод</i>
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 D. Две окружности

Имя входного файла: *стандартный ввод*  
Имя выходного файла: *стандартный вывод*  
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

<i>стандартный ввод</i>
2 0 0 2 4 0 2 0 0 1 100 100 1
<i>стандартный вывод</i>
There are only 1 of them.... 2.0000000000000000 0.0000000000000000  There are no points!!!

## Задача Е. Принадлежность точки отрезку

Имя входного файла: *стандартный ввод*  
Имя выходного файла: *стандартный вывод*  
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 шесть 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

<i>стандартный ввод</i>	<i>стандартный вывод</i>
3 3 1 2 5 4	YES

## Задача F. Площадь треугольника

Имя входного файла: *стандартный ввод*  
Имя выходного файла: *стандартный вывод*  
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

<i>стандартный ввод</i>	<i>стандартный вывод</i>
0 0 0 1 1 1	0.5
-2 -2 0 0 2 3	1

## Задача G. Площадь выпуклого многоугольника

Имя входного файла: *стандартный ввод*  
Имя выходного файла: *стандартный вывод*  
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

<i>стандартный ввод</i>	<i>стандартный вывод</i>
3 0 0 0 1 2 0	1
4 0 0 0 1 1 1 1 0	1

## Задача Н. Треугольник и точка

Имя входного файла: *стандартный ввод*  
Имя выходного файла: *стандартный вывод*  
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

<i>стандартный ввод</i>	<i>стандартный вывод</i>
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 I. Точка в многоугольнике

Имя входного файла: *стандартный ввод*  
Имя выходного файла: *стандартный вывод*  
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

<i>стандартный ввод</i>	<i>стандартный вывод</i>
3 0 0 1 0 0 1 1 1	NO

## Задача J. Подобие треугольников

Имя входного файла: *стандартный ввод*  
Имя выходного файла: *стандартный вывод*  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Витя прочитал в книжке, что *подобие* — это такое преобразование, которое изменяет длины всех отрезков в фиксированное вещественное число раз.

Кроме того, в книжке пишут, что *треугольник* — это произвольный набор из трёх точек!

А два треугольника называются подобными, если существует преобразование подобия, переводящее первый треугольник во второй, либо второй в первый, с сохранением порядка вершин.

Требуется определить, подобны ли два треугольника в Витином понимании.

### Формат входных данных

Первая строка ввода содержит число тестовых наборов входных данных  $C$ , которое должно быть не более 1000. Каждый из наборов состоит из описания двух треугольников. Треугольник задаётся тремя своими вершинами. Все координаты — целые числа, по абсолютной величине не превосходящие 10 000. Все числа разделяются произвольным количеством пробелов и переводов строки.

### Формат выходных данных

Для каждого из наборов входных данных выведите в отдельной строке слово «YES», если два треугольника подобны, иначе выведите «NO».

### Пример

<i>стандартный ввод</i>	<i>стандартный вывод</i>
2	YES
0 0 1 1 2 0	NO
0 1 0 0 1 0	
0 0 1 1 2 0	
0 1 1 0 0 0	

## Задача К. Квадрат

Имя входного файла: *стандартный ввод*  
Имя выходного файла: *стандартный вывод*  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

На уроке геометрии при обсуждении задачи о квадратуре круга семиклассник 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

<i>стандартный ввод</i>	<i>стандартный вывод</i>
1 0 0 1	1 1 0 0

## Задача L. Космическая полиция

Имя входного файла: *стандартный ввод*  
Имя выходного файла: *стандартный вывод*  
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 abordaj.

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 abordaj, 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

<i>стандартный ввод</i>	<i>стандартный вывод</i>
1 1 2 1 4 4	3.0000000000
0 0 0 1 0 -2	-1

## Задача М. Тип треугольника

Имя входного файла: *стандартный ввод*  
Имя выходного файла: *стандартный вывод*  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Даны координаты трёх точек на плоскости. Нарисуем треугольник с вершинами в этих точках. Определите тип этого треугольника: остроугольный, прямоугольный, тупоугольный или вырожденный.

- Если все углы треугольника строго меньше 90 градусов, это остроугольный треугольник.
- Если один из углов треугольника равен 90 градусам, это прямоугольный треугольник.
- Если один из углов треугольника строго больше 90 градусов, это тупоугольный треугольник.
- Если все три точки лежат на одной прямой, критерии об углах не применяются, и треугольник считается вырожденным.

### Формат входных данных

В первой строке записаны два числа  $x_1$  и  $y_1$  — координаты первой точки. Во второй строке записаны два числа  $x_2$  и  $y_2$  — координаты второй точки. В третьей строке записаны два числа  $x_3$  и  $y_3$  — координаты третьей точки. Все числа целые и лежат в пределах от  $-100$  до  $+100$ .

### Формат выходных данных

Выведите одно слово:

- «acute», если треугольник остроугольный,
- «right», если треугольник прямоугольный,
- «obtuse», если треугольник тупоугольный,
- «degenerate», если треугольник вырожденный.

### Примеры

<i>стандартный ввод</i>	<i>стандартный вывод</i>
0 0 3 4 6 0	acute
6 0 3 1 0 0	obtuse

## Задача N. Классификация четырёхугольников

Имя входного файла: *стандартный ввод*  
Имя выходного файла: *стандартный вывод*  
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

<i>стандартный ввод</i>	<i>стандартный вывод</i>
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 О. Центр

Имя входного файла: *стандартный ввод*  
Имя выходного файла: *стандартный вывод*  
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:

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

Например,  $\rho_1(A, B) = |x_2 - x_1| + |y_2 - y_1|$  — это манхэттенское расстояние (количество перекрёстков на пути между заданными точками), а  $\rho_2(A, B) = \sqrt{|x_2 - x_1|^2 + |y_2 - y_1|^2}$  — 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

<i>стандартный ввод</i>	<i>стандартный вывод</i>
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 Р. Выпуклая оболочка

Имя входного файла: *стандартный ввод*  
Имя выходного файла: *стандартный вывод*  
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

<i>стандартный ввод</i>	<i>стандартный вывод</i>
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 Q. Сортировка на плоскости

Имя входного файла:	<i>стандартный ввод</i>
Имя выходного файла:	<i>стандартный вывод</i>
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

*Это интерактивная задача.*

Есть  $n$  векторов на плоскости, ненулевых и попарно неколлинеарных. Вектор с номером  $i$  идёт из начала координат в точку  $(x_i, y_i)$ . Но эти координаты мы вам не скажем.

Вместо этого вы можете задавать вопросы следующего вида: «Верно ли, что векторы  $i$  и  $j$  образуют правую пару?» Формально векторы образуют правую пару, если  $x_i \cdot y_j > x_j \cdot y_i$ . Геометрически правая пара означает, что, если мы стоим в начале координат и смотрим на конец вектора  $i$ , то, чтобы как можно скорее повернуться к концу вектора  $j$ , следует поворачиваться против часовой стрелки.

Нужно, задавая жюри вопросы, прийти к одному из двух выводов:

1. Все векторы лежат в одной полуплоскости, граница которой проходит через начало координат. Тогда нужно отсортировать их: вывести такой набор различных индексов  $i_1, i_2, \dots, i_n$ , что для любых  $p < q$  векторы  $i_p$  и  $i_q$  образуют правую пару.
2. Нет такой полуплоскости, граница которой проходит через начало координат и в которой лежат все векторы. Тогда нужно доказать это: вывести такую последовательность различных индексов  $i_1, i_2, \dots, i_k$ , что каждый вектор, кроме последнего, образует правую пару со следующим, а последний ( $i_k$ ) образует правую пару с первым ( $i_1$ ).

### Протокол взаимодействия

Сначала вашей программе подаётся в отдельной строке число  $n$  — количество векторов ( $1 \leq n \leq 500$ ). Векторы в каждом тесте зафиксированы заранее, но держатся в секрете.

Затем вы можете делать следующее:

1. Спросить у жюри: «верно ли, что векторы  $i$  и  $j$  образуют правую пару?»

Для этого ваша программа должна вывести строку следующего вида: «?  $i$   $j$ ». Индексы должны быть корректными:  $1 \leq i, j \leq n$ .

В ответ программа жюри выдаст в отдельной строке число: 1, если ответ «да», и 0, если ответ «нет».

Чтобы предотвратить буферизацию вывода, после каждого выведенного вопроса следует вставить команду очистки буфера вывода: например, это может быть `fflush (stdout)` в C или C++, `System.out.flush ()` в Java, `flush (output)` в Pascal или `sys.stdout.flush ()` в Python.

Вы можете задать не более 20 000 вопросов.

2. Вывести ответ. В этом случае ваша программа должна вывести две строки.

Если все векторы лежат в одной полуплоскости, первая строка должна иметь вид «! YES», а во второй должны быть выведены через пробел  $n$  различных чисел от 1 до  $n$ : индексы векторов в порядке сортировки.

Если такой полуплоскости нет, выведите в первой строке «! NO». В начале второй строки выведите  $k$  — число векторов в доказательстве, а затем выведите  $k$  различных чисел от 1 до  $n$  — само доказательство. Если возможных доказательств несколько, выведите любое из них.

После вывода ответа ваша программа должна сразу корректно завершить работу.

## Примеры

<i>стандартный ввод</i>	<i>стандартный вывод</i>
3 0 1 0	? 1 3 ? ? 3 2 ? ? 2 1 ! ! YES 3 1 2
3 1 0 0	? 1 2 ? ? ? 3 2 ? ? 1 3 ! ! NO 3 1 2 3

## Задача R. Футбол

Имя входного файла:	<i>стандартный ввод</i>
Имя выходного файла:	<i>стандартный вывод</i>
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Вы с друзьями играете в футбол. Вы находитесь в точке  $(0, 0)$  на бесконечном плоском поле и хотите отдать пас своему сокоманднику, стоящему в точке  $(x_t, y_t)$ . К несчастью, на поле также есть игрок противоположной команды в точке  $(x_o, y_o)$ , который хочет этот пас перехватить.

Формально, игрок получает мяч тогда и только тогда, когда его позиция в точности совпадает с позицией мяча. Оба игрока могут двигаться со скоростью  $s_p$ , а Вы можете придать мячу скорость  $s_b$ . Обратите внимание, что Вы не можете двигаться с мячом, а также, что игроки начинают двигаться только после удара по мячу. Для простоты будем считать, что трение мяча о поле мало, но при этом, если никто из игроков не сможет перехватить мяч во время его движения, то трение все равно остановит мяч где-то далеко-далеко, так что игроки смогут его догнать.

Ваша задача заключается в том, чтобы выяснить, можно ли отдать пас своему сокоманднику. Выведите «+» (без кавычек), если Вы можете отдать такой пас, что сокомандник получит мяч строго раньше оппонента, «0», если это невозможно, но при этом можно отдать такой пас, что игроки смогут настичуть мяч одновременно, и «-» иначе.

### Формат входных данных

В первой строке записано одно число  $n$  ( $1 \leq n \leq 10^5$ ): количество наборов входных данных, для которых надо решить задачу. Каждая из последующих  $n$  строк содержит шесть целых чисел  $x_t, y_t, x_o, y_o, s_b, s_p$  ( $-75 \leq x_t, y_t, x_o, y_o \leq 75$ ;  $1 \leq s_b, s_p \leq 75$ ): координаты сокомандника, координаты соперника и скорости мяча и игроков, соответственно. Гарантируются, что позиции всех игроков на поле попарно различны.

### Формат выходных данных

В единственной строке выведите  $n$  ответов без пробелов.

### Пример

<i>стандартный ввод</i>	<i>стандартный вывод</i>
3 1 0 2 0 1 1 0 3 0 -1 1 2 75 0 2 0 1 10	+0-

## Задача S. Пары монстров

Имя входного файла: *стандартный ввод*  
Имя выходного файла: *стандартный вывод*  
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

<i>стандартный ввод</i>	<i>стандартный вывод</i>
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 Т. Четырёхугольники

Имя входного файла: *стандартный ввод*  
Имя выходного файла: *стандартный вывод*  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

На плоскости дано  $N$  точек. Никакие две точки не совпадают, никакие три не лежат на одной прямой. Выясните, сколько существует различных выпуклых четырёхугольников с вершинами в этих точках. Два четырёхугольника считаются различными, если множества их вершин не совпадают.

### Формат входных данных

В первой строке входных данных задано целое число  $N$  — количество точек ( $4 \leq N \leq 1500$ ). Каждая из следующих  $N$  строк содержит два целых числа  $X_i$  и  $Y_i$ , разделённые пробелом — координаты очередной точки. Все координаты не превосходят по модулю  $10^8$ . Гарантируется, что никакие две точки не совпадают и никакие три не лежат на одной прямой.

### Формат выходных данных

Выведите одно целое число — количество различных выпуклых четырёхугольников с вершинами в этих точках.

### Примеры

<i>стандартный ввод</i>	<i>стандартный вывод</i>
4 -1 -1 -1 1 1 1 1 -1	1
5 0 0 5 0 3 4 0 5 5 5	3

### Пояснения к примерам

Иллюстрация ко второму примеру:

