Задача A. Площадь выпуклого многоугольника

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

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: $|x_i|, |y_i| \leqslant 10\,000$, $3 \leqslant N \leqslant 100$.

Формат выходных данных

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

Примеры

стандартный ввод	**стандартный вывод**
3 0 0 0 1 2 0	1
4 0 0 0 1 1 1 1 0	1

Задача B. Точка в многоугольнике

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

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 ($3 \leqslant N \leqslant 100\,000$) 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

стандартный ввод	*стандартный вывод*
3 0 0 1 0 0 1 1 1	NO

Задача C. Тип треугольника

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

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

стандартный ввод	***стандартный вывод***

`0 0`
`3 4`
`6 0`
	`acute`

`6 0`
`3 1`
`0 0`
	`obtuse`

Задача D. Классификация четырёхугольников

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

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 ($-100 \leqslant x, y \leqslant 100$). 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

стандартный ввод	стандартный вывод

`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 E. Части плоскости

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

Даны N точек на плоскости. Проведём прямые через каждую пару точек. На сколько частей эти прямые делят плоскость?

Формат входных данных

В первой строке задано число N — количество точек ($2 \leqslant N \leqslant 10$). Следующие N строк содержат по два числа X_i и Y_i каждая через пробел — координаты i-й точки ($-100 \leqslant X_i, Y_i \leqslant 100$). Никакие две данные точки не совпадают, никакие три не лежат на одной прямой. Все числа во входных данных целые.

Формат выходных данных

В первой строке выведите P — количество частей, на которые полученные прямые делят плоскость.

Примеры

стандартный ввод	**стандартный вывод**
4 0 0 0 1 1 0 1 1	16
3 1 5 2 3 -8 4	7

Задача F. Ареал

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

Специалисты, исследующие дагестанских туров, пришли к выводу, что в течение года каждый отдельно взятый тур мигрирует в пределах некоторой области, которую можно описать как многоугольник Ψ. Летний ареал туров также представляет собой многоугольник Φ. Полный ареал туров теперь можно определить как *сумму Минковского* $\Psi \oplus \Phi$.

Суммой Минковского множеств точек (векторов) Ψ и Φ называется множество точек $c = a + b$, где $a \in \Psi$ и $b \in \Phi$.

Ваша задача — вычислить площадь полного ареала туров.

Формат входных данных

Входные данные состоят из описаний двух выпуклых многоугольников — Ψ и Φ. Описание каждого из них начинается со строки, содержащей число вершин N ($2 < N \leqslant 10^5$). Затем следуют N строк, каждая содержит координаты очередной вершины (x_i, y_i), разделённые пробелами. Вершины задаются в порядке обхода против часовой стрелки, начиная с любой из них; все координаты целые и не превосходят по модулю 10^8. Последовательные три вершины не могут лежать на одной прямой.

Формат выходных данных

Выведите одно вещественное число с шестью или более верными знаками после десятичной точки — площадь искомого ареала.

Пример

стандартный ввод	стандартный вывод
4 0 0 1 0 1 1 0 1 4 0 1 -1 0 0 -1 1 0	7.000000

Задача G. Пары монстров

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

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 ($1 \leqslant n \leqslant 200\,000$). Следующие n строк описывают монстров. Каждая строка содержит два целых числа a_i и b_i, разделённых пробелом — уродливость и отвратительность i-го монстра ($|a_i|, |b_i| \leqslant 10^9$).

Формат выходных данных

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

Примеры

стандартный ввод	стандартный вывод
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 H. Четырёхугольники

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

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 ($4 \leqslant N \leqslant 1500$). 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

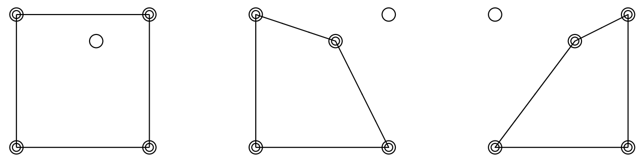
Выведите одно целое число — количество различных выпуклых четырёхугольников с вершинами в этих точках.

Примеры

стандартный ввод	стандартный вывод
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:



Задача I. Преобразования плоскости

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

Рассмотрим последовательность преобразований плоскости, каждое из которых состоит в том, чтобы повернуть и масштабировать плоскость относительно фиксированной точки так, чтобы один фиксированный вектор, направленный из этой точки, переходил в другой вектор.

Даны точка на плоскости и последовательнось преобразований указанного вида. В какую точку плоскости перейдёт данная точка после последовательного применения всех преобразований?

Формат входных данных

В первой строке записаны через пробел три целых числа n, x и y ($1 \leqslant n \leqslant 100$, $|x|, |y| \leqslant 100$). Следующие n строк содержат описания преобразований в порядке их применения; в i-й из этих строк записаны через пробел шесть целых чисел $p_x^{(i)}$, $p_y^{(i)}$, $u_x^{(i)}$, $u_y^{(i)}$, $v_x^{(i)}$ и $v_y^{(i)}$, означающих, что плоскость поворачивается и масштабируется относительно точки $(p_x^{(i)}, p_y^{(i)})$ так, что вектор $u^{(i)} = (u_x^{(i)}, u_y^{(i)})$ переходит в вектор $v^{(i)} = (v_x^{(i)}, v_y^{(i)})$ (другими словами, точка $(p_x^{(i)} + u_x^{(i)}, p_y^{(i)} + u_y^{(i)})$ переходит в точку $(p_x^{(i)} + v_x^{(i)}, p_y^{(i)} + v_y^{(i)})$) ($|p_x^{(i)}|$, $|p_y^{(i)}|$, $|u_x^{(i)}|$, $|u_y^{(i)}|$, $|v_x^{(i)}|$, $|v_y^{(i)}| \leqslant 100$, векторы $u^{(i)}$ и $v^{(i)}$ — ненулевые).

Формат выходных данных

В первой строке выведите два числа через пробел — координаты точки, в которую переместится точка (x, y) в результате данных преобразований. Координаты должны иметь относительную или абсолютную погрешность не более 10^{-9}. Допускается экспоненциальная форма вывода ответа.

Примеры

стандартный ввод
1 2 0
0 0 1 0 0 2
стандартный вывод
0 4

стандартный ввод
2 0 0
3 5 -1 1 1 -1
5 3 1 -1 -1 1
стандартный вывод
4.0000 -4.0000

стандартный ввод
1 1 3
0 0 5 1 1 -1
стандартный вывод
8.461538461538e-001 2.307692307692e-001

Задача J. Футбол

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

Вы с друзьями играете в футбол. Вы находитесь в точке $(0, 0)$ на бесконечном плоском поле и хотите отдать пас своему сокомандинку, 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 настигнуть мяч одновременно, и «-» иначе.

Формат входных данных

В первой строке записано одно число n ($1 \leqslant n \leqslant 10^5$): количество наборов входных данных, для которых надо решить задачу. Каждая из последующих n строк содержит шесть целых чисел $x_t, y_t, x_o, y_o, s_b, s_p$ ($-75 \leqslant x_t, y_t, x_o, y_o \leqslant 75$; $1 \leqslant s_b, s_p \leqslant 75$): 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

стандартный ввод	стандартный вывод
3 1 0 2 0 1 1 0 3 0 -1 1 2 75 0 2 0 1 10	+0-