

## Задача А. Кто ходит в гости по утрам?

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

Пятачок и Винни-Пух каждое утро ходят пить чай в гости к Кролику. Естественно, самым коротким путем.

К сожалению, однажды Винни-Пуху пришла в голову идея вырыть ловушку для Слонопотама. Самое обидное, что они с Пятачком ее даже вырыли. Поэтому теперь каждое утро, идя в гости к Кролику, они боятся в нее провалиться.

Напишите программу, которая посчитает длину самого короткого безопасного пути от домика Винни-Пуха до домика Кролика.

Ловушка для Слонопотама представляет собой яму абсолютно круглой формы. Путь является безопасным, если он не проходит по ловушке (но может проходить по ее границе).

### Формат входных данных

Во входном файле записаны сначала координаты домика Винни-Пуха  $x_a, y_a$ , затем — координаты домика Кролика  $x_b, y_b$ , а затем — координаты центра и радиус ловушки  $x_c, y_c, r_c$ . Все координаты — целые числа из диапазона от  $-32000$  до  $32000$ . Радиус ловушки — натуральное число, не превышающее  $32000$ .

Домики Винни-Пуха и Кролика не могут находиться внутри ловушки, но могут находиться на ее границе.

### Формат выходных данных

Выведите в выходной файл одно число — длину самого короткого безопасного пути от домика Винни-Пуха до домика Кролика с точностью не менее 4 знаков после запятой.

### Примеры

| стандартный ввод  | стандартный вывод |
|-------------------|-------------------|
| 0 0 10 0<br>5 5 1 | 10                |
| 3 4 4 4<br>0 0 5  | 1                 |

## Задача В. Парк

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

В столице Байтленда есть огороженный парк, территория которого представляет собой прямоугольник. Деревья и посетители в парке представлены в виде кругов.

В парке четыре входа, по одному в каждом углу (1 — внизу слева, 2 — внизу справа, 3 — вверху справа, 4 — вверху слева). Посетители могут входить и выходить из парка только через входы. Посетители могут входить и выходить из парка, когда они касаются обеих сторон угла соответствующего входа. Посетители могут свободно перемещаться по парку, но они не могут перекрывать деревья и заборы.

Ваша задача — рассчитать для каждого посетителя, учитывая вход, в который они войдут в парк, через какие выходы они могут выйти из парка.

### Формат входных данных

Первая строка ввода содержит два целых числа  $n$  и  $m$  ( $1 \leq n \leq 2000$   $1 \leq m \leq 100\,000$ ) — количество деревьев в парке и количество посетителей.

Вторая строка ввода содержит два целых числа  $w$  и  $h$  ( $1 \leq w, h \leq 10^9$ ) — ширину и высоту парковой зоны. Нижний левый угол имеет координаты  $(0, 0)$ , а верхний правый угол имеет координаты  $(w, h)$ .

После этого есть  $n$  строк, которые описывают деревья. Каждая строка содержит три целых числа  $x$ ,  $y$  и  $r$ , что означает, что центр  $i$ -го дерева  $(x, y)$  и его радиус  $r$ . Деревья не перекрывают друг друга или забор.

Наконец, есть  $m$  строк, которые описывают посетителей. Каждая строка содержит два целых числа  $r$  и  $e$  — радиус посетителя и вход, в который они войдут в парк.

Кроме того, ни одно дерево не перекрывает квадратную область  $2k \times 2k$  в каждом углу, где  $k$  — радиус наибольшего посетителя.

### Формат выходных данных

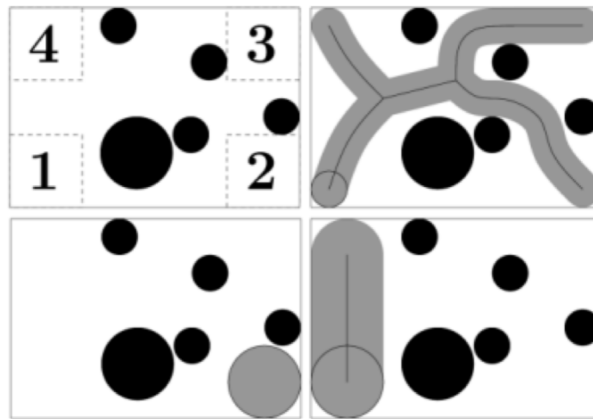
Вы должны вывести для каждого посетителя одну строку, содержащую входы, через которые они могут выйти из парка, в отсортированном порядке без пробелов между ними.

### Пример

| стандартный ввод | стандартный вывод |
|------------------|-------------------|
| 5 3              | 1234              |
| 16 11            | 2                 |
| 11 8 1           | 14                |
| 6 10 1           |                   |
| 7 3 2            |                   |
| 10 4 1           |                   |
| 15 5 1           |                   |
| 1 1              |                   |
| 2 2              |                   |
| 2 1              |                   |

### Замечание

На следующем рисунке показаны зоны входа и возможные маршруты для каждого посетителя:



## Задача С. Теодор Рузвельт

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 64 мегабайта

«Теодор Рузвельт» — флагман военно-морского флота Кукуляндии. Заклятые враги кукуляндцев, флатландцы, решили уничтожить его. Они узнали, что «Теодор Рузвельт» представляет собой выпуклый многоугольник из  $n$  вершин и узнали его координаты. Затем они выпустили  $m$  баллистических ракет и определили координаты точек, где эти ракеты взорвались. По расчётам штаба флатландцев, «Теодор Рузвельт» будет уничтожен, если в него попадёт хотя бы  $k$  ракет. Вычислите, удалось ли флатландцам уничтожить корабль.

### Формат входных данных

В первой строке через пробел записаны целые числа  $n, m, k$  ( $3 \leq n \leq 10^5, 0 \leq k \leq m \leq 10^5$ ). В последующих  $n$  строках записаны координаты вершин многоугольника в порядке обхода против часовой стрелки. В следующих  $m$  строках записаны координаты точек. Гарантируется, что все координаты — целые числа, не превосходящие по модулю  $10^9$ .

### Формат выходных данных

Выведите «YES», если в многоугольнике или на его границе лежит по крайней мере  $k$  точек, и «NO» в противном случае.

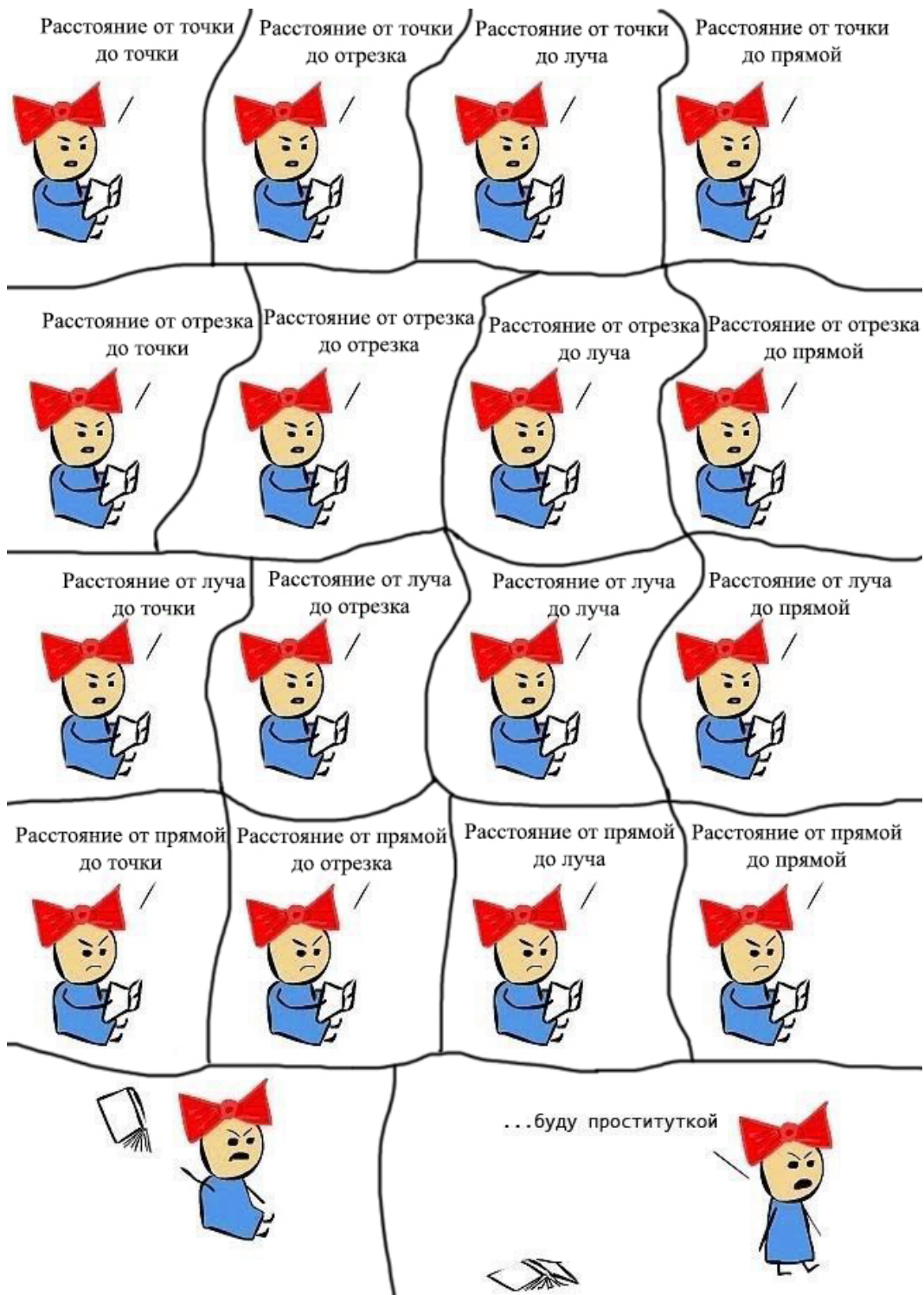
### Пример

| стандартный ввод  | стандартный вывод |
|---|-------------------|
| 5 4 2<br>1 -1<br>1 2<br>0 4<br>-1 2<br>-1 -1<br>-2 -1<br>1 -1<br>0 1<br>2 3 | YES               |

## Задача D. 16

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

Даны 4 точки A, B, C, D.  
Посчитайте:



Даны 4 точки A, B, C, D.

Посчитайте:

- Расстояние от точки A до точки C.
- Расстояние от точки A до отрезка CD.
- Расстояние от точки A до луча CD.
- Расстояние от точки A до прямой CD.
- Расстояние от отрезка AB до точки C.
- Расстояние от отрезка AB до отрезка CD.
- Расстояние от отрезка AB до луча CD.
- Расстояние от отрезка AB до прямой CD.
- Расстояние от луча AB до точки C.
- Расстояние от луча AB до отрезка CD.
- Расстояние от луча AB до луча CD.
- Расстояние от луча AB до прямой CD.
- Расстояние от прямой AB до точки C.
- Расстояние от прямой AB до отрезка CD.
- Расстояние от прямой AB до луча CD.
- Расстояние от прямой AB до прямой CD.

### Формат входных данных

Даны координаты четырех точек, по одной точке в строке:  $X_a, Y_a, X_b, Y_b, X_c, Y_c, X_d, Y_d$ . Все числа целые, по модулю не превосходят 10000.

### Формат выходных данных

Выведите 16 чисел по одному в строке. Числа в ответе должны быть выданы с точностью не менее 6 знаков после десятичной точки.

### Пример

| стандартный ввод | стандартный вывод |
|------------------|-------------------|
| 1 2              | 5.6568542495      |
| 7 1              | 5.6000000000      |
| 5 6              | 5.6000000000      |
| 8 2              | 5.6000000000      |
|                  | 4.6031716446      |
|                  | 1.4142135624      |
|                  | 1.4000000000      |
|                  | 1.4000000000      |
|                  | 4.6031716446      |
|                  | 1.1507929111      |
|                  | 0.0000000000      |
|                  | 0.0000000000      |
|                  | 4.6031716446      |
|                  | 1.1507929111      |
|                  | 0.0000000000      |
|                  | 0.0000000000      |

## Задача E. Целочисленные точки

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

Многоугольник (не обязательно выпуклый) на плоскости задан координатами своих вершин. Требуется подсчитать количество точек с целочисленными координатами, лежащих внутри него (но не на его границе).

### Формат входных данных

В первой строке вводится  $n$  ( $1 \leq n \leq 1000$ ) — число вершин многоугольника. В последующих  $n$  строках идут координаты  $(x_i, y_i)$  вершин многоугольника в порядке обхода по часовой стрелке.  $x_i$  и  $y_i$  — целые числа, по модулю не превосходящие  $10^6$ .

### Формат выходных данных

Выведите одно число — количество искомых точек.

### Пример

| стандартный ввод              | стандартный вывод |
|-------------------------------|-------------------|
| 4<br>1 1<br>1 2<br>2 2<br>2 1 | 0                 |

## Задача F. Две окружности-1

|                         |                   |
|-------------------------|-------------------|
| Имя входного файла:     | стандартный ввод  |
| Имя выходного файла:    | стандартный вывод |
| Ограничение по времени: | 1 секунда         |
| Ограничение по памяти:  | 256 мегабайт      |

Юный футболист Митя обнаружил на школьном футбольном поле две различные окружности, нарисованные едва заметной белой краской. Вспомнив истории о загадочных кругах на полях, он отметил эти окружности с помощью небольших камушков. Митя разложил на поле  $n$  камушков так, чтобы каждый из них находился на одной из окружностей или даже на их пересечении, если эти окружности пересекаются. Получилось так, что на каждой окружности размещался хотя бы один камушек. Обладая великолепным глазомером, Митя расположил камушки на окружностях абсолютно точно, без какой-либо погрешности.

На следующий день пошел дождь, краска стерлась, и нарисованные окружности исчезли, но все камушки остались на своих местах. Теперь Мите очень нужно найти доказательство необычного явления, свидетелем которого он был, то есть, восстановить окружности.

Требуется написать программу, которая по координатам камушков на поле находит вариант размещения их на двух несовпадающих окружностях.

### Формат входных данных

Первая строка входного файла содержит целое число  $n$  — количество размещенных Митей камушков на поле ( $2 \leq n \leq 2000$ ). Последующие  $n$  строк содержат целочисленные координаты  $(x_i, y_i)$  камушков — по одной паре координат, разделенных пробелом, в каждой строке ( $-10^6 \leq x_i, y_i \leq 10^6$ ). Никакие два камушка не размещаются в одной точке.

Гарантируется, что ответ для заданного набора камушков существует.

### Формат выходных данных

Выходной файл должен содержать две строки. Первая строка должна содержать количество камушков, принадлежащих первой окружности, а потом последовательность номеров всех этих камушков, вторая строка — количество камушков, принадлежащих второй окружности, и последовательность номеров всех таких камушков.

Каждый камушек должен встречаться хотя бы в одной из двух последовательностей. Если камушек встречается в обеих последовательностях, то это обозначает, что он находится на пересечении окружностей. Считается, что камушки пронумерованы от 1 до  $n$  в порядке их следования во входных данных.

Нумерация окружностей не имеет значения, то есть выводить две последовательности можно в любом порядке. Числа в последовательностях можно также выводить в произвольном порядке. Каждая из последовательностей должна содержать не менее одного числа. Все числа в строках должны быть разделены пробелами.

Если вариантов расположения окружностей несколько, можно выбрать любой из них.



## Примеры

| стандартный ввод   | стандартный вывод      |
|--|------------------------|
| 7<br>1 -1<br>0 0<br>1 1<br>3 1<br>3 -1<br>2 0<br>4 0           | 4 1 2 3 6<br>4 4 5 7 6 |
| 5<br>-1000000 0<br>0 1000000<br>1000000 0<br>0 -1000000<br>0 0 | 4 1 2 3 4<br>1 5       |

## Задача G. Извилистая ломаная

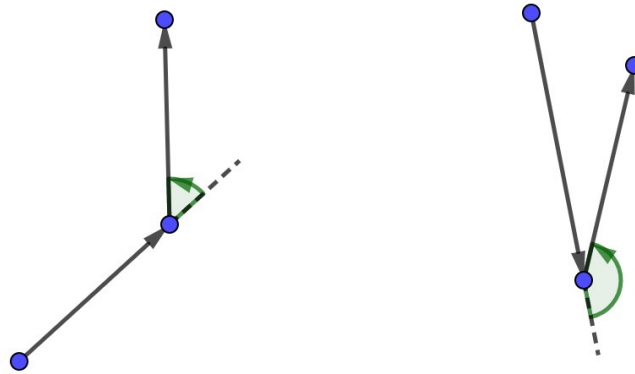
|                         |                   |
|-------------------------|-------------------|
| Имя входного файла:     | стандартный ввод  |
| Имя выходного файла:    | стандартный вывод |
| Ограничение по времени: | 1 секунда         |
| Ограничение по памяти:  | 256 мегабайт      |

У Васи есть  $n$  различных точек  $A_1, A_2, \dots, A_n$  на плоскости. Никакие три из них не лежат на одной прямой. Он хочет расположить их в некотором порядке  $A_{p_1}, A_{p_2}, \dots, A_{p_n}$ , где  $p_1, p_2, \dots, p_n$  — это некоторая перестановка чисел от 1 до  $n$ .

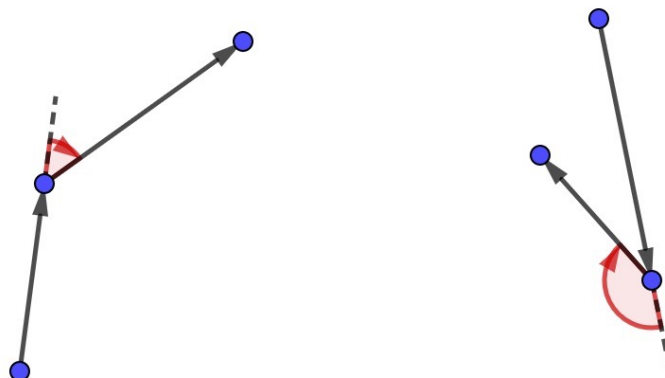
Сделав так, он нарисует ориентированную ломаную на этих вершинах, проведя направленные отрезки из каждой точки в следующую в выбранном порядке точку. То есть для всех  $1 \leq i \leq n-1$  он проведет направленный отрезок из точки  $A_{p_i}$  в точку  $A_{p_{i+1}}$ . Он хочет, чтобы получившаяся ломаная удовлетворяла 2-м условиям:

- она будет несамопересекающейся, то есть любые 2 отрезка, которые не являются соседними, не имеют общих точек.
- она будет извилистой.

У Васи есть строка  $s$ , состоящая из  $(n-2)$ -х символов “L” или “R”. Будем называть направленную ломаную извилистой, если её  $i$ -й поворот будет налево, если  $s_i = \text{“L”}$  и направо, если  $s_i = \text{“R”}$ . Более формально:  $i$ -й поворот ломаной будет в точке  $A_{p_{i+1}}$ , в ней направленный отрезок из точки  $A_{p_i}$  в точку  $A_{p_{i+1}}$  поменяется на направленный отрезок из точки  $A_{p_{i+1}}$  в точку  $A_{p_{i+2}}$ . Обозначим вектор  $\vec{v}_1 = \overrightarrow{A_{p_i}A_{p_{i+1}}}$  и вектор  $\vec{v}_2 = \overrightarrow{A_{p_{i+1}}A_{p_{i+2}}}$ . Тогда если для того, чтобы повернуть вектор  $\vec{v}_1$  на наименьший возможный угол, чтобы его направление совпало с направлением вектора  $\vec{v}_2$  надо сделать поворот против часовой стрелки, то будем говорить, что  $i$ -й поворот налево, а иначе направо. Для лучшего понимания посмотрите картинку, на которых изображены различные варианты поворотов:



На этой картинке изображены повороты налево



На этой картинке изображены повороты направо

Вам даны координаты точек  $A_1, A_2, \dots, A_n$  на плоскости и строка  $s$ . Найдите перестановку  $p_1, p_2, \dots, p_n$  чисел от 1 до  $n$ , такую что ломаная, которую нарисует Вася, будет удовлетворять двум заданным условиям.

### Формат входных данных

В первой строке написано одно целое число  $n$  — количество точек ( $3 \leq n \leq 2000$ ). В следующих  $n$  строках написаны по два целых числа  $x_i$  и  $y_i$ , разделённые пробелом — координаты точки  $A_i$  на плоскости ( $-10^9 \leq x_i, y_i \leq 10^9$ ). В последней строке написана строка  $s$  из символов “L” и “R” длины  $(n - 2)$ . Гарантируется, что все точки различны и никакие три точки не лежат на одной прямой.

### Формат выходных данных

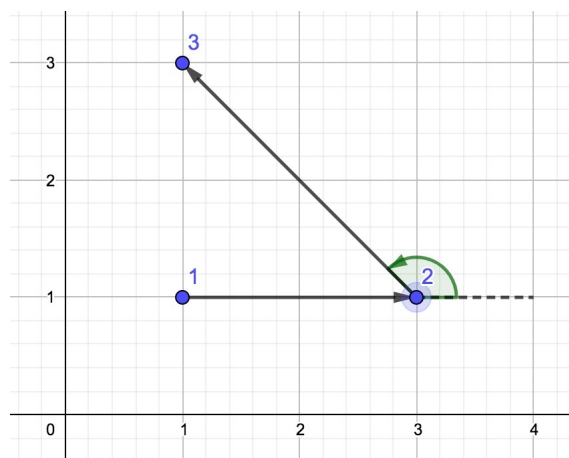
Если подходящей перестановки не существует выведите  $-1$ . Иначе выведите  $n$  чисел  $p_1, p_2, \dots, p_n$  — найденную перестановку ( $1 \leq p_i \leq n$  и все  $p_1, p_2, \dots, p_n$  различны). Если подходящих перестановок несколько, выведите любую.

### Примеры

| стандартный ввод                                       | стандартный вывод |
|--|-------------------|
| 3<br>1 1<br>3 1<br>1 3<br>L                            | 1 2 3             |
| 6<br>1 0<br>0 1<br>0 2<br>-1 0<br>-1 -1<br>2 1<br>RLLR | 5 4 1 6 2 3       |

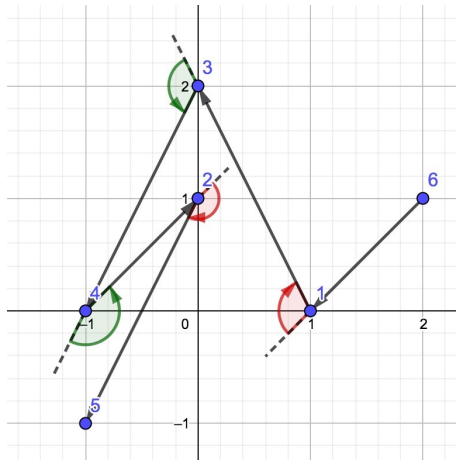
### Замечание

Вот картинка, изображающая ломаную из 1 теста:



Как мы видим, эта ломаная несамопересекающаяся, а также извилистая, так как поворот в точке 2 налево.

Вот картинка, изображающая ломаную из 2 теста:



## Задача Н. Две окружности-2

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

Найдите точки пересечения двух окружностей.

### Формат входных данных

На вход программе подаются шесть целых чисел  $x_a, y_a, r_a, x_b, y_b, r_b$  ( $-10^6 \leq x_a, y_a, r_a, x_b, y_b, r_b \leq 10^6$ ), задающие координаты центра и радиусы двух окружностей.

### Формат выходных данных

В случае, если количество общих точек окружностей конечно, в первой строке вывести одно число  $K$ , равное этому количеству, далее в  $K$  строках координаты самих точек с точностью не менее 10 знаков после запятой. Если указанных точек бесконечно много, вывести единственное число «3».

### Примеры

| стандартный ввод | стандартный вывод                  |
|------------------|------------------------------------|
| 3 4 5 11 4 2     | 0                                  |
| 3 4 5 10 4 2     | 1<br>8.000000000000 4.000000000000 |

## Задача I. Выпуклые оболочки

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

Выпуклая оболочка множества точек плоскости — наименьший выпуклый многоугольник, содержащий эти точки.

Вам дано  $n$  точек на плоскости. Произвольно одна из них выбирается и удаляется.

Найдите среднее число вершин выпуклой оболочки результирующего множества. В этой задаче считайте, что если выпуклая оболочка — отрезок, то в ней две вершины. Если же она — невырожденный многоугольник, то все углы при вершинах строго меньше 180 градусов.

### Формат входных данных

В первой строке содержится единственное число  $n$  ( $3 \leq n \leq 200\,000$  — количество точек во множестве). В последующих  $n$  строках заданы пары чисел, не превышающих по модулю  $10^9$  — координаты точек. Никакие две точки не совпадают.

### Формат выходных данных

Выведите среднее число вершин в выпуклой оболочке множества без одной точки в виде несократимой дроби  $p/q$ .

### Пример

| стандартный ввод                     | стандартный вывод |
|--------------------------------------|-------------------|
| 5<br>0 0<br>0 4<br>4 0<br>3 3<br>4 4 | 17/5              |

## Задача J. Платные дороги

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

Мэр одного большого города решил ввести плату за проезд по шоссе, проходящим в районе города, чтобы снизить объем транзитного транспорта. В районе города проходит  $n$  шоссе.

Но руководство области, в которой расположен город, воспротивилось планам мэра. Действительно — дальнбойщики представляют собой неплохой источник доходов для большого количества кафе и гостиниц в небольших городках.

В результате решили, что плата будет введена только на шоссе, которые проходят через город.

В городе используется развитая система метрополитена, всего в городе есть  $m$  станций метро. Решено было, что шоссе проходит через город, если либо одна из станций метро расположена непосредственно на шоссе, либо есть хотя бы одна станция с каждой стороны от шоссе.

Помогите теперь мэру определить, какие шоссе проходят через город.

### Формат входных данных

Первая строка входного файла содержит два целых числа:  $n$  и  $m$  — количество шоссе и количество станций метро, соответственно ( $1 \leq n, m \leq 100\,000$ ).

Следующие  $n$  строк описывают шоссе. Каждое шоссе описывается тремя целыми числами  $a$ ,  $b$  и  $c$  и представляет собой прямую на плоскости, задаваемую уравнением  $ax + by + c = 0$  ( $|a|, |b|, |c| \leq 10^9$ ).

Следующие  $m$  строк входного файла описывают станции метро. Каждая станция описывается двумя целыми числами  $x$  и  $y$  и представляет собой точку на плоскости с координатами  $(x, y)$  ( $|x|, |y| \leq 10^9$ ).

### Формат выходных данных

Первая строка выходного файла должна содержать одно целое число — количество шоссе, которые проходят через город. Вторая строка должна содержать номера этих шоссе в возрастающем порядке. Шоссе нумеруются от 1 до  $n$  в порядке, в котором они описаны во входном файле.

### Примеры

| стандартный ввод   | стандартный вывод |
|--|-------------------|
| 4 2<br>0 1 0<br>1 0 1<br>1 1 0<br>1 1 -1<br>0 0<br>2 0   | 3<br>1 3 4        |
| 12 1<br>0 1 0<br>0 1 1<br>0 1 -1<br>1 0 1<br>1 0 -1<br>1 0 0<br>1 1 0<br>1 1 1<br>1 1 -1<br>1 -1 0<br>1 -1 1<br>1 -1 -1<br>0 0 | 4<br>1 6 7 10     |

## Задача К. Волшебный лес

|                         |                   |
|-------------------------|-------------------|
| Имя входного файла:     | стандартный ввод  |
| Имя выходного файла:    | стандартный вывод |
| Ограничение по времени: | 1 секунда         |
| Ограничение по памяти:  | 256 мегабайт      |

В волшебном лесу растут  $n$  деревьев. На плане леса они изображены точками (диаметром деревьев можно пренебречь). Территорией леса считается наименьший по площади выпуклый многоугольник (возможно, вырожденный), содержащий в себе все деревья.

Отважный путешественник и писатель Ручкин однажды решил на отчаянный поступок — он совершил путешествие в этот лес. После этого он описал свое путешествие в книге. В частности, в книге описаны все деревья леса в том порядке, в каком они встречались Ручкину (каждое дерево описано ровно один раз).

Художник Кистин решил нарисовать иллюстрацию для этой книги. Для этого он приехал и остановился в деревне недалеко от волшебного леса. Теперь он хочет выбрать точку, с которой он будет рисовать иллюстрацию. Кистин очень боится заходить в волшебный лес, поэтому хочет найти точку для рисования обязательно за пределами леса (в том числе, она не может находиться на границе леса).

Он решил нарисовать весь лес: он хочет взять длинный-длинный холст, и зарисовать весь лес справа налево, от самой правой точки леса до самой левой. При этом деревья леса должны на картине идти справа налево ровно в том же порядке, в котором они описаны в книге. Естественно, никакое дерево не должно быть заслонено другим деревом (т.е. на отрезке между Кистиным и деревом не может быть других деревьев).

Помогите ему: напишите программу, которая по координатам деревьев волшебного леса в том порядке, в каком они описаны у Ручкина, поможет Кистину выбрать точку, из которой деревья видны в требуемом порядке.

### Формат входных данных

Задано число  $n$  — количество деревьев в лесу ( $1 \leq n \leq 100000$ ). Далее перечислено  $n$  пар чисел, задающих координаты деревьев в том порядке, в каком они описаны в книге Ручкина. Все координаты — целые числа, не превосходящие по абсолютной величине  $10^5$ . Гарантируется, что никакие два дерева не растут в одной точке.

### Формат выходных данных

Если подобрать точку для Кистина возможно, выведите сообщение «Possible», а в следующей строке — два вещественных числа: координаты точки. Координаты выведенной точки не должны превышать  $10^{15}$  по абсолютной величине. Если подобрать точку с указанными ограничениями не удастся, выведите сообщение «Impossible».

При проверке ответа для случая «Possible» он будет считаться верным, если на расстоянии менее  $10^{-5}$  от выведенной точки будет существовать точка, удовлетворяющая условию.



## Примеры

| стандартный ввод              | стандартный вывод |
|-------------------------------|-------------------|
| 3<br>0 0<br>1 2<br>2 1        | Possible<br>1 4   |
| 3<br>1 0<br>2 0<br>3 0        | Possible<br>1 1   |
| 3<br>1 0<br>3 0<br>2 0        | Impossible        |
| 4<br>0 0<br>2 3<br>4 2<br>3 1 | Impossible        |
| 4<br>0 0<br>4 0<br>2 2<br>4 4 | Possible<br>-2 2  |