

## Задача А. Футбол

Имя входного файла: `football.in`  
Имя выходного файла: `football.out`  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

На футбольном поле размером  $x \times y$  находятся  $n$  футболистов. Они уже очень устали и стоят на месте, но ждут, куда упадет мяч, чтобы побежать к нему. Футболист бежит к мячу в том случае, если мяч упал к этому футболисту ближе, чем к любому другому футболисту. Требуется определить для каждого футболиста границы зоны, при попадании в которую он побежит к мячу, если известно, что она представляет собой многоугольник.

### Формат входных данных

В первой строке входного файла заданы три целых числа  $x$ ,  $y$  и  $n$  ( $2 \leq x, y \leq 10^5$ ,  $1 \leq n \leq 1000$ ). Следующие  $n$  строк содержат целые координаты футболистов  $x_i y_i$  ( $0 < x_i < x$ ,  $0 < y_i < y$ ). Никакие два футболиста не стоят в одной точке.

### Формат выходных данных

В выходной файл выведите  $n$  строк. В каждой из строк первое число — количество вершин зоны  $k_i$ , далее  $k_i$  чисел — координаты вершин  $x_{ij} y_{ij}$  в порядке обхода против часовой стрелки, начиная с самой нижней из самых левых вершин зоны. Вещественные числа выводите с максимальной точностью.

### Примеры

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

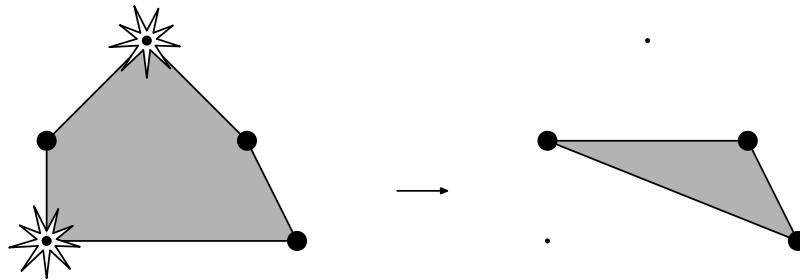
## Задача В. База в джунглях

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

Глубоко в джунглях расположена военная база. Она окружена  $n$  сторожевыми башнями с ультразвуковыми генераторами. В этой задаче сторожевые башни считаются точками на плоскости.

Сторожевые башни генерируют ультразвуковое поле с помощью которого защищают все объекты, расположенные строго внутри выпуклой оболочки сторожевых башен. Не существует сторожевой башни строго внутри выпуклой оболочки и никакие три сторожевые башни не находятся на одной прямой.

Враг может уничтожить несколько башен. Если такое происходит, защищаемая область уменьшается до выпуклой оболочки оставшихся башен.



Командир базы хочет построить штаб внутри защищаемой области. Чтобы максимизировать ее безопасность, он хочет максимизировать минимальное количество башен, который должен уничтожить враг, чтобы оставить штаб без защиты.

### Формат входных данных

Первая строка входного файла содержит одно число  $n$  ( $3 \leq n \leq 50\,000$ ) — количество сторожевых башен. Следующие  $n$  строк содержат декартовы координаты сторожевых башен, по одной паре в строке. Все координаты являются целыми числами и не превосходят по модулю  $10^6$ .

Башни описаны в порядке обхода их выпуклой оболочки по часовой стрелке.

### Формат выходных данных

Выведите в выходной файл количество сторожевых башен, которые должен уничтожить враг, чтобы оставить штаб без защиты, если штаб будет расположен оптимально.

### Примеры

<code>jungle.in</code>	<code>jungle.out</code>
3 0 0 50 50 60 10	1
5 0 0 0 10 10 20 20 10 25 0	2

## Задача С. Волшебный лес

Имя входного файла:	<code>forest.in</code>
Имя выходного файла:	<code>forest.out</code>
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

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 ( $1 \leq N \leq 100\,000$ ). 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

forest.in	forest.out
3 0 0 1 2 2 1	Possible 0.99999839817756997000 2.00000987087457640000
3 1 0 2 0 3 0	Possible 1 1
3 1 0 3 0 2 0	Impossible
4 0 0 2 3 4 2 3 1	Impossible
4 0 0 4 0 2 2 4 4	Possible -0.00000382683432365090 0.00000923879532511287

## Задача D. Триатлон

Имя входного файла: `tri.in`  
Имя выходного файла: `tri.out`  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

Триатлон — атлетическое соревнование, состоящее из трех последовательных этапов, которые суммарно должны быть пройдены как можно быстрее. Первый этап — плавание, второй — велогонка, а третий — бег.

Известна скорость каждого участника для каждого этапа. Жюри соревнования может выбрать длину каждого этапа любым образом так, чтобы каждый этап не был нулевой длины. В результате, иногда жюри может выбрать длины этапов так, чтобы определенный участник стал победителем соревнования.

### Формат входных данных

В первой строке входного файла задано целое число  $N$  — количество участников соревнования ( $1 \leq N \leq 100$ ). Следующие  $N$  строк содержат по 3 целых числа  $V, U$  и  $W$  ( $1 \leq U, V, W \leq 10000$ ), определяющие скорость очередного участника на этапах соревнования.

### Формат выходных данных

Для каждого участника выведите одну строку, содержащую слово *Yes* в случае если жюри может подобрать длины этапов так, чтобы этот участник стал победителем (т.е. этот участник — единственный, кто первый дошел до финиша), или слово *No* в противном случае.

### Примеры

<code>tri.in</code>	<code>tri.out</code>
9	Yes
10 2 6	Yes
10 7 3	Yes
5 6 7	No
3 2 7	No
6 2 6	No
3 5 7	Yes
8 4 6	No
10 4 2	Yes
1 8 7	