

## Содержание

Задача А. Выпуклая оболочка [0.5 sec, 256 mb]	2
Задача В. Футбол [0.5 sec, 256 mb]	3

---

### Задача А. Выпуклая оболочка [0.5 sec, 256 mb]

Дано  $N$  точек на плоскости.

Нужно построить их выпуклую оболочку.

Гарантируется, что выпуклая оболочка не вырождена.

#### Формат входных данных

В первой строке число  $N$  ( $3 \leq N \leq 10^5$ ). Следующие  $N$  строк содержат пары целых чисел  $x$  и  $y$  ( $-10^9 \leq x, y \leq 10^9$ ) — точки.

Будьте аккуратны! Точки произвольны. Бывают совпадающие, бывают лежащие на одной прямой в большом количестве.

#### Формат выходных данных

В первой строке выведите  $N$  — число вершин выпуклой оболочки. Следующие  $N$  строк должны содержать координаты вершин в порядке обхода. Никакие три подряд идущие точки не должны лежать на одной прямой.

#### Пример

stdin	stdout
5	4
0 0	0 0
2 0	2 0
0 2	2 2
1 1	0 2
2 2	

### Задача В. Футбол [0.5 sec, 256 mb]

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.

#### Пример

stdin	stdout
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	