

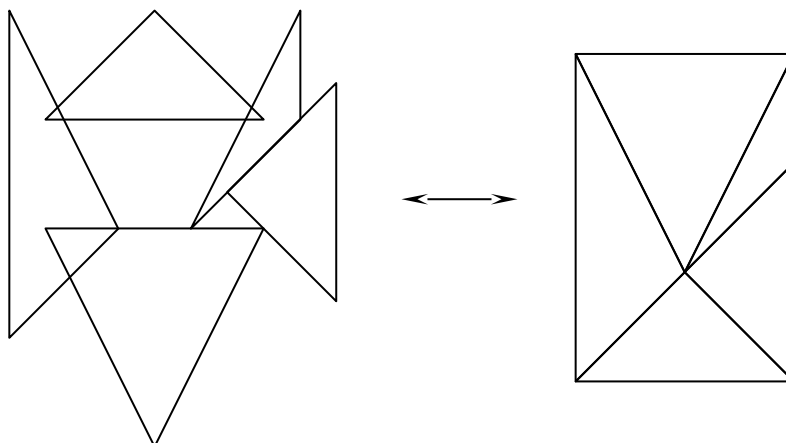
Задача 5. Стекло

Имя входного файла: `glass.in`
Имя выходного файла: `glass.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 32 мегабайта
Максимальная оценка: 100 баллов

На столе расположено несколько треугольных осколков стекла. Были выдвинуты следующие предположения:

- все эти осколки принадлежали одному прямоугольному стеклу, которое исходно лежало так, что его стороны были параллельны краям стола;
- осколки получены в результате удара в некоторую точку стекла, не лежащую на его границе;
- в момент удара одна из вершин каждого осколка находилась в точке удара;
- некоторые осколки, возможно, были сдвинуты;
- стекло может быть восстановлено параллельным переносом осколков, то есть ни один из осколков не был повернут.

Напишите программу, помогающую при этих предположениях полностью восстановить прямоугольное стекло из **всех** имеющихся осколков или определяющую, что это сделать невозможно.



Формат входных данных

В первой строке входного файла записано число N — количество осколков ($1 \leq N \leq 2000$).

Введем систему координат таким образом, чтобы оси координат были параллельны сторонам стола. В каждой из следующих N строк записано по три пары чисел, задающих текущие координаты вершин осколков. Все координаты — целые числа, по модулю не превосходящие 10000. Все осколки имеют ненулевую площадь.

Формат выходных данных

Если стекло возможно восстановить при указанных предположениях, то в N строках выходного файла выведите координаты вершин осколков после восстановления стекла. Стекло должно быть восстановлено таким образом, чтобы координаты всех его углов были неотрицательны, и один из его углов располагался в начале координат. Перечисление осколков и их вершин должно соответствовать их порядку следования во входном файле.

В случае нескольких решений выведите любое из них.

В случае невозможности восстановить стекло в выходной файл выведите сообщение NO.

Продолжение задачи на следующей странице

Примеры

glass.in	glass.out
5 0 0 3 3 0 9 1 3 7 3 4 -3 1 6 7 6 4 9 5 3 8 6 8 9 6 4 9 1 9 7	0 0 3 3 0 9 0 9 6 9 3 3 0 0 6 0 3 3 3 3 6 6 6 9 3 3 6 0 6 6
4 0 0 20 -20 20 20 0 0 -20 20 -20 -20 0 0 -20 20 20 20 -1 0 20 -20 -20 -20	NO