UVa Online Judge

1206 Boundary Points

The convex hull of a set of points Q on a plane is the smallest convex polygon P for which each point in Q is in the boundary or inside P.

"The problem of finding convex hulls finds its practical applications in pattern recognition, image processing, statistics and geographic information systems or GIS. It also serves as a tool, a building block for a number of other computational-geometric algorithms. For example, consider the problem of finding the diameter of a set of points, which is the pair of points a maximum distance apart. The diameter will always be the distance between two points on the convex hull.

For planar objects, i.e., lying in the plane, the convex hull may be easily visualized by imagining an elastic band stretched open to encompass the given objects; when released, it will assume the shape of the required convex hull. It may seem natural to generalise this picture to higher dimensions by imagining the objects enveloped in a sort of idealized unpressurized elastic membrane or balloon under tension. However, the equilibrium (minimum-energy) surface in this case may not be the convex hull– parts of the resulting surface may have negative curvature, like a saddle surface. For the case of points in 3-dimensional space, if a rigid wire is first placed between each pair of points, then the balloon will spring back under tension to take the form of the convex hull of the points." (*from Wikipedia*)

Write a program that outputs the convex polygon P for a set of points Q.

Input

The set of points in Q are given in one line and the program should be able to read any number of lines where each line is one set of points in Q. Each point is of the format (x, y) where $x, y \in R$.

Output

Each test case will be displayed in one line of output. The first point and the last point of the convex hull must match. Any sequence of points defining the convex hull will be considered valid.

Sample Input

```
(-2,1) (-1,-2) (-1,1) (-1,2) (-1,3) (0,0) (1,-1) (1,1) (2,-2) (2,1) (3,2)
```

Sample Output

```
(-1,-2) (2,-2) (3,2) (-1,3) (-2, 1) (-1,-2)
```