Back To The Polygon

A simple polygon is a polygon that does not overlap with itself. A diagonal of a simple polygon is a segment within the polygon that connects two non-consecutive vertices. A triangulation of a simple polygon of N edges is the drawing of exactly N – 3 diagonals that do not touch each other anywhere, with the possible exception of their endpoints. A triangulation divides the polygon in exactly N – 2 triangles that do not overlap and only touch each other along their edges.

In this problem, you are given the triangulation of a simple polygon, which means, the set of triangles in which a polygon was divided. From them, you need to reconstruct the original polygon.

Input

The input contains several test cases, each one described in several lines. The first line of each test case contains an integer N (3 ≤ N ≤ 500), the number of edges of the original polygon. Each of the next N – 2 lines describes one triangle in the triangulation of the polygon. Each triangle is given by six integers X1 , Y1 , X2 , Y2 , X3 and Y3 separated by single spaces, where Xi
and Yi are the coordinates in the XY plane of the i-th vertex of the triangle (–1000 ≤ Xi , Yi ≤ 1000). The triangles and their vertices are not given in any specific order. The last line of the input contains a single –1 and should not be processed as a test case.

Output

For each test case output a single line with 2N integers separated by single spaces. These integers must represent the coordinates in the XY plane of the vertices of the original polygon, in clockwise order. To make the output unique, the first vertex to be listed is the one with the smallest X coordinate, and if there are many of those, the one with the smallest Y coordinate among them.

Example

Input:

```
5
0 0 10 9 10 0
10 9 0 9 0 0
10 9 0 9 5 13
3
0 1 1 1 1 0
10
-1 -2 2 -2 2 -1
-1 -2 0 -1 -2 3
-2 3 2 3 1 2
0 -1 2 1 1 2
-1 -2 2 -1 0 -1
2 1 0 -1 2 0
2 3 2 2 1 2
0 -1 -2 3 1 2
-1
```

Output:

```
0 0 0 9 5 13 10 9 10 0
0 1 1 1 1 0
-2 3 2 3 2 2 1 2 2 1 2 0 0 -1 2 -1 2 -2 -1 -2
```