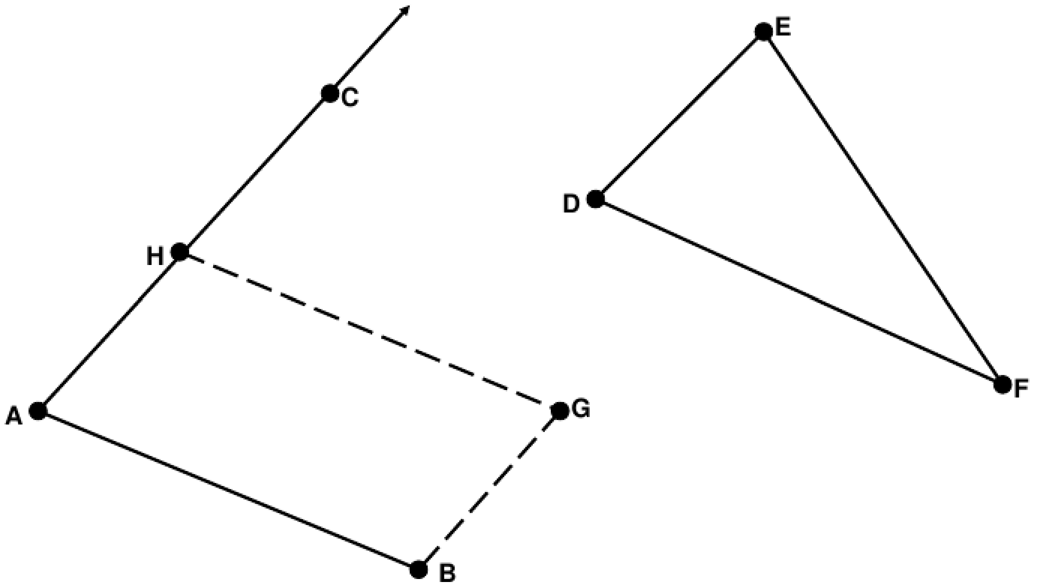
In one of his notebooks, Euclid gave a complex procedure for solving the following problem. With computers, perhaps there is an easier way.

In a 2D plane, consider a line segment AB, another point C which is not collinear with AB, and a triangle DEF. The goal is to find points G and H such that:

- H is on the ray AC (it may be closer to A than C or further away, but angle CAB is the same as angle HAB)
- ABGH is a parallelogram (AB is parallel to GH, AH is parallel to BG)
- The area of parallelogram ABGH is the same as the area of triangle DEF



Input

Input consists of multiple datasets. Each dataset will consist of twelve real numbers, with no more than 3 decimal places each, on a single line. Those numbers will represent the x and y coordinates of points A through F, as follows:

x_A y_A x_B y_B x_C y_C x_D y_D x_E y_E x_F y_F

Points A, B and C are guaranteed to **not** be collinear. Likewise, D, E and F are also guaranteed to be non-collinear. Every number is guaranteed to be in the range from $-1000.0 \ldots 1000.0$ inclusive.

End of the input will be a line with twelve zero values (`0.0`).

Output

For each input set, print a single line with four floating point numbers. These represent points G and H, like this:

x_G y_G x_H y_H

Print all values to a precision of 3 decimal places (rounded, NOT truncated). Print a single space between numbers.

Sample Input

```
0 0 5 0 0 5 3 2 7 2 0 4
1.3 2.6 12.1 4.5 8.1 13.7 2.2 0.1 9.8 6.6 1.9 6.7
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
```

Sample Output

```
5.000 0.800 0.000 0.800
13.756 7.204 2.956 5.304
```