

Problem H

Higgs Boson

Source file name: `higgs.c`, `higgs.cpp` or `higgs.java`

It's been 100 years since the detection of the first Higgs boson and now particle physics is a mainstream subject in all high schools. Obviously, kids love the fact that they can create tiny black holes using only their portable particle accelerators and show off to their friends and colleagues. Although the creation of big black holes that could swallow the whole planet is possible even with these portable particle accelerators, the devices are programmed to only throw particles when this undesirable side effect is impossible.

Your granddaughter is trying to create her own black holes with a portable accelerator kit, which is composed of two small particle accelerators that throw, each one, a boson-sized particle. Both particles are thrown at the same time, and a black hole appears when the particles collide. However, your granddaughter doesn't know how much time she'll have to wait before this happens. Fortunately, each accelerator can predict the particle's trajectory, showing four integer values into its display, called A , B , C and D . Each value can be replaced into the following equations:

$$\begin{aligned}r &= At + B \\ \theta &= Ct + D\end{aligned}$$

in order to determine the trajectory of the particle, in polar coordinates. The radius (r) is represented in distance units and the angle (θ) in degrees. The time (t) is given in time units and it is always a rational value which can be represented by an irreducible fraction. Your granddaughter knows that in polar coordinates a point has infinite representations. In general, the point (r, θ) can be represented as $(r, \theta \pm k \times 360^\circ)$ or $(-r, \theta \pm (2k + 1) \times 180^\circ)$, where k is any integer. Besides, the origin ($r = 0$) can be represented as $(0, \theta)$ for any θ .

Using these parameters informed by each particle accelerator, your granddaughter wants to determine whether the particles will eventually collide and, if they do, the time when they will collide. After the first collision it is impossible to predict the particle's trajectory, therefore, only the first possible collision should be considered.

Although your granddaughter is really intelligent and has a deep knowledge of particle physics, she does not know how to program computers and is looking for some notes in her grandfather's (or grandmother's) ICPC notebook (don't forget, she is *your* granddaughter!). Fortunately for you, there is a note on your notebook which says that you wrote that code during the 2008 ICPC South America Regional Contest (or, to be more specific, *this* contest).

Input

The input consists of several test cases, one per line. Each test case contains eight integer numbers separated by single spaces, $A_1, B_1, C_1, D_1, A_2, B_2, C_2, D_2$ ($-10^4 \leq A_1, B_1, C_1, D_1, A_2, B_2, C_2, D_2 \leq 10^4$). The first four input values (A_1, B_1, C_1, D_1) correspond to the four parameters displayed by the first portable particle accelerator and the following input values (A_2, B_2, C_2, D_2) correspond to the four parameters displayed by the second portable particle accelerator when both particles are thrown. The end of the input is represented by $A_1 = B_1 = C_1 = D_1 = A_2 =$

$B_2 = C_2 = D_2 = 0$, which should not be processed as a test case, since these are the values displayed by the particle accelerators when a big black hole would be created if the particles were thrown. Although the end of input is represented by a line with eight zeroes, note that the number zero is a possible input value.

The input must be read from standard input.

Output

For each test case, your program must output a line containing two non-negative integers t_a and t_b separated by a single space. If there is no possibility of collision, $t_a = t_b = 0$, otherwise, t_a/t_b must be an irreducible fraction representing the earliest collision time. Even if the fraction results in an integer value, you still must output the number 1 as the denominator (see samples below).

The output must be written to standard output.

Sample input	Output for the sample input
1 1 180 0 2 0 180 360	1 1
10 10 360 0 -24 18 180 72	0 0
5 5 180 0 -12 9 10 40	4 17
-9 5 5 180 2 5 5 180	0 1
0 0 0 0 0 0 0 0	