Problem 10

uPhone Two-Point Zoom

8 points

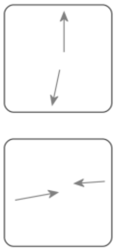
JAVA: program name must be prob10.class
C /C++ program name must be: prob10.exe

Task Description

Pear Computer Corporation is developing a product called the uPhone. One of the features of the uPhone is the ability to zoom and rotate photos in and out using two-finger pinch and expand. Pear Computer Corporation is convinced people will pay big bucks for this kind of feature. The touch-screen hardware has already been developed, now they just need a little software magic. That's where you come in.
Write a program to convert two-point touch-screen gestures into parameters that control movement, zoom and rotation.

Program Input

Every line of input contains four pairs of positive integers, each pair representing x and y screen coordinates. Each finger touching the screen produces two pairs of x-y coordinates: the start point and the end point, hence, four total pairs. The numbers are: start1-x, start1-y, end1-x, end1-y, start2-x, start2-y, end2-x, and end2-y. Luckily for you, code already exists for filtering invalid gestures, so all input will result in valid operations. The program must exit when it reads a line that contains all zeroes.



```
18 41 96 57 184 123 141 99
58 24 42 36 148 176 164 98
97 81 97 146 81 48 65 16
0 0 0 0 0 0 0 0
```

Program Output

The program must print four numbers for each line of input. The first two numbers represent relative x and y motion of the photo. Relative motion must be computed from the difference of the centers of the end points minus the centers of the start points. The third number represents the zoom factor expressed as a real number. To compute this zoom factor, divide the distance between the end points by the distance between the start points. The distance *d* between two points *(x1,y1)* and *(x2,y2)* is found using the following formula:

$$d^2 = (x_2 - x_1)^2 + (y_2 - y_1)^2$$

The fourth number indicates counter-clockwise rotation expressed in radians. For this computation you need to know how to compute the angle between two lines. For two lines with slopes m_1 and m_2, the angle a between the lines is given by the formula:

$$\tan a = \frac{m_2 - m_1}{1 + m_1 m_2}$$

For this program you will assume there are no vertical lines (thus no infinite slope *m*) and that all rotations are within the range of +/- 90 degrees (1.571 radians). The zoom factor and rotation angle must be accurate within a margin of error of +/- 0.001. Larger errors will be judged as incorrect.

```
17 -4 0.332 0.292
0 -33 0.775 -0.566
-8 17 3.651 0.210
```

* In Java, C, and C++ you'll use the **atan** function to calculate the arctangent. So if tan(a) = b, then a=atan(b). In Pascal this same function is called **ArcTan**.