```
 1  /* Problem 5--Olympic Hide And Seek Final
 2     My first attempt was to compute the three angles of the triangle and
 3     apply the Law of Sines twice.  However, there were lots of cases to
 4     check to get the correct angles...and it was easier just to compute
 5     the coordinates of the locators and draw vectors out from them. */
 6
 7  import java.io.*;
 8  import java.util.*;
 9
10  public class prob5 {
11
12   private static Scanner in;
13   private static PrintWriter out;
14   private static int cs;
15   private static double side, a1, a2, a3;
16
17   public static void main (String[] args) throws Exception {
18
19    in = new Scanner (new File ("prob5.in"));
20    out = new PrintWriter ("prob5.out");
21    cs = 1;
22    while (true) {
23     side = in.nextDouble(); //distance to second locator
24     a1 = in.nextDouble(); //angle to second locator
25     a2 = in.nextDouble(); //angle from first locator to emitter
26     a3 = in.nextDouble(); //angle from second locator to emitter
27     if (side==0 && a1==0 && a2==0 && a3==0) break;
28     double x1=0, y1 = 0; //First locator at (0,).
29     double x2 = x1 + side*Math.cos(a1*Math.PI/180),
30            y2 = y1 + side*Math.sin(a1*Math.PI/180); //second locator
31     double s1 = Math.sin (a2*Math.PI/180), //sins and coss so that I
32            c1 = Math.cos (a2*Math.PI/180); //don't have to recompute them
33     double s2 = Math.sin (a3*Math.PI/180),
34            c2 = Math.cos (a3*Math.PI/180);
35     double t = (s1*(x2-x1)-c1*(y2-y1))/(s2*c1-s1*c2); //Distance to
36     double s; //second emitter
37     if (Math.abs(c1) > 1e-5) s = (t*c2+x2-x1)/c1;//Two formulas for first
38     else s = (t*s2+y2-y1)/s1; //emitter.  sin and cos can't both be zero.
39     out.printf ("Case %d: %.1f miles from the first receiver and %.1f "+
40                 "miles from the second!\r\n\r\n",cs++,s,t);
41    }
42    in.close ();
43    out.close ();
44   }
45  }
46
```