/* Problem 5--Olympic Hide And Seek Final
My first attempt was to compute the three angles of the triangle and
apply the Law of Sines twice. However, there were lots of cases to
check to get the correct angles...and it was easier just to compute
the coordinates of the locators and draw vectors out from them. */

import java.io.*;
import java.util.*;

public class prob5 {
  private static Scanner in;
  private static PrintWriter out;
  private static int cs;
  private static double side, a1, a2, a3;

  public static void main (String[] args) throws Exception {
    in = new Scanner (new File("prob5.in"));
    out = new PrintWriter("prob5.out");
    cs = 1;
    while (true) {
      side = in.nextDouble(); //distance to second locator
      a1 = in.nextDouble(); //angle to second locator
      a2 = in.nextDouble(); //angle from first locator to emitter
      a3 = in.nextDouble(); //angle from second locator to emitter
      if (side==0 && a1==0 && a2==0 && a3==0) break;
      double x1=0, y1 = 0; //First locator at (0,).
      double x2 = x1 + side*Math.cos(a1*Math.PI/180),
                y2 = y1 + side*Math.sin(a1*Math.PI/180); //second locator
      double s1 = Math.sin (a2*Math.PI/180), //sins and coss so that I
                    c1 = Math.cos (a2*Math.PI/180); //don't have to recompute them
      double s2 = Math.sin (a3*Math.PI/180),
                    c2 = Math.cos (a3*Math.PI/180);
      double t = (s1*(x2-x1)-c1*(y2-y1))/(s2*c1-s1*c2); //Distance to
                  double s; //second emitter
      if (Math.abs(c1) > 1e-5) s = (t*c2+x2-x1)/c1; //Two formulas for first
                  else s = (t*s2+y2-y1)/s1; //emitter. sin and cos can't both be zero.
      out.printf("Case %d: %.1f miles from the first receiver and %.1f +
                   "miles from the second!\r\n\r\n",cs++,s,t);
    }
    in.close();
    out.close();
  }
}