```c
/*    Copyright (c) 2017 Nathan Christian (nchristi@iu.edu)
      Lab 3: Program 3: Prints circles and a line connecting the two
circles
      Intended for output with winfig to create a .fig file
      Intended to read coordinates from a file (not user input)*/

#include <stdio.h>
#include <math.h>
#include "fig.h"

double distance (double x11, double y11, double x22, double y22);
//distance function from distance.c

int main()
{
      double x0, y0, r0, x1, y1, r1, theta;
      int p1[2], p2[2], count =6;

      fig_open(stdout); //function from fig.h for drawing figures;
      fig_userinfo(stdout, 1, "ECE263L3P1", "Nathan Christian",
"6/12/2017"); //prints my user info to .fig file

      while (count==6) {

                  count=scanf("%lf %lf %lf %lf %lf %lf", &x0, &y0, &r0,
&x1, &y1, &r1); //input center coordinates and radius of two circles


                  theta=atan((fabs((y1-y0)/(x1-x0)))); //finds the angle
used for trig functions below

                  if    (distance(x0,y0,x1,y1)-r1-r0<= 0) {
                              fig_circle(stdout, fig_convert(x0),
fig_convert(y0), fig_convert(r0), "color=black;line_style=3");
                              fig_circle(stdout, fig_convert(x1),
fig_convert(y1), fig_convert(r1), "color=black;line_style=3");
                  }   //if (circles overlap) draw dashes


                  else { //finds trig coordinates and prints thick solid
for lines and circles

                              fig_circle(stdout, fig_convert(x0),
fig_convert(y0), fig_convert(r0), "color=red;thickness=5");
                              fig_circle(stdout, fig_convert(x1),
fig_convert(y1), fig_convert(r1), "color=red;thickness=5"); //prints solid
thick red circles

                              if (x0<x1) {
                                    p1[0]=fig_convert(x0+r0*cos(theta));
                                    p1[1]=fig_convert(x1-r1*cos(theta));
                              } //trig functions for determining two
closest points of the two circles
```

```
                              if (y0>y1) {
                                    p2[0]=fig_convert(y0-r0*sin(theta));
                                    p2[1]=fig_convert(y1+r1*sin(theta));
                              } //trig function for determining two closest
points of the two circles

                              if (x0>x1) {
                                    p1[0]=fig_convert(x0-r0*cos(theta));
                                    p1[1]=fig_convert(x1+r1*cos(theta));
                              } //trig function for determining two closest
points of the two circles

                              if (y0<y1) {
                              p2[0]=fig_convert(y0+r0*sin(theta));
                              p2[1]=fig_convert(y1-r1*sin(theta));
                              } //trig function for determining two closest
points of the two circles

                                    fig_polyline(stdout, p1, p2, 2,
"color=blue;thickness=5"); // prints thick blue connecting lines
                  } // end if (circles do not overlap)
      } //end while

      return;
}
```