```
//YingYang
#include <math.h>
#include<iostream>
#include "glut.h"
#define SCREENWIDTH 640
#define SCREENHEIGHT 480
struct Point
{
    int x, y;
};
void myInit(void)
{
    glClearColor(0.5,0.5,0.5,0.0);
    glColor3f(1.0,0.0f,1.0f);
    glPointSize(2.0);
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    gluOrtho2D(0.0,SCREENWIDTH,0.0,SCREENHEIGHT);

}
void drawArc(Point center, GLfloat radius, GLfloat startingAngle, GLfloat endingAngle, GLfloat r,
GLfloat g, GLfloat b)
{
  const float Pi = 4*atan(1.0);
  Point prev, current;
  glColor3f(r,g,b);
  glBegin(GL_LINES);
        glVertex2f(center.x, center.y);
        glVertex2f(center.x + radius * cos((startingAngle * Pi)/180), center.y + radius *
sin((startingAngle * Pi)/180));
  glEnd();
  for(GLfloat angle = (startingAngle * Pi)/180; angle <= (endingAngle * Pi)/180; angle += Pi/180)
  {
      current.x = center.x + radius * cos(angle);
      current.y = center.y + radius * sin(angle);
      if(angle != (startingAngle * Pi)/180 )
      {
            glBegin(GL_LINES);
                glVertex2f(prev.x, prev.y);
                glVertex2f(current.x, current.y);
            glEnd();
            glBegin(GL_POLYGON);
                glVertex2f(center.x, center.y);
                glVertex2f(prev.x, prev.y);
                glVertex2f(current.x, current.y);
            glEnd();
            glFlush();
      }
      prev.x = current.x;
      prev.y = current.y;
  }
```

```
 glBegin(GL_LINES);
        glVertex2f(prev.x, prev.y);
        glVertex2f(center.x, center.y);
 glEnd();
}
void YingYang()
{
    Point center, leftCenter, rightCenter;
    // your code here

    //Draw upper half circle with black color

    //Draw lower half circle with black color

    //Draw left lower half of circle with subRadius1 with black color

    //Draw left upper half of circle with subRadius1 with white color

    //Draw a circle to left with subRadius2 and color white

    //Draw a circle to right with subRadius2 and color Black

}
void myDisplay()
{
    glClear(GL_COLOR_BUFFER_BIT);
    glRecti(100,20,540,460);
    YingYang();
}

void main(int argc, char **argv)
{

    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
    glutInitWindowSize(SCREENWIDTH,SCREENHEIGHT);
    glutInitWindowPosition(100, 150);
    glutCreateWindow("Ying-Yang");
    glutDisplayFunc(myDisplay);
    myInit();
    glutMainLoop();
}
```