```
#include<iostream>
#include<GL/glut.h>
using namespace std;
int w = 600;
int h = 400;
void coordinate();
void myInit();
void MyDisplay();
void bresenhamCircle(int x,int y,int r);
int main(int a,char **v) {
glutInit(&a,v);
glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
glutInitWindowPosition(0,0);
glutInitWindowSize(h,w);
glutCreateWindow("Bresenham Circle");
myInit();
glutDisplayFunc(MyDisplay);
glutMainLoop();
return 0;
}
void myInit() {
glPointSize(3.0);
glClearColor(1.0,1.0,1.0,0);
glColor3f(1.0,0.0,0.0);
gluOrtho2D(-w/2,w/2,-h/2,h/2);
}
void coordinate() {
glBegin(GL_LINES);
glVertex2d(-w/2,0);
glVertex2d(w/2,0);
glVertex2d(0,-h/2);
```

```
glVertex2d(0,h/2);
glEnd();
glFlush();
}
void MyDisplay() {
glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
coordinate();

bresenhamCircle(0,0,48);
bresenhamCircle(50,80,48);
bresenhamCircle(-50,80,48);
bresenhamCircle(-50,-80,48);
bresenhamCircle(50,-80,48);
bresenhamCircle(100,0,48);
bresenhamCircle(-100,0,48);
glFlush();
}
void bresenhamCircle(int x,int y,int r) {
int xi = 0;
int yi = r;
int pk = 3-2*r;
int pi = pk;
while(xi<=yi) {
glBegin(GL_POINTS);
glVertex2f(x+xi,y+yi);
glVertex2f(x+yi,y+xi);
glVertex2f(x+yi,y- xi);
glVertex2f(x+xi,y- yi);
glVertex2f(x-xi,y-yi);
glVertex2f(x-yi,y-xi);
glVertex2f(x-yi,y+xi);
```

```
glVertex2f(x-xi,y+yi);
glEnd();
glFlush();
if(pi<0) {
xi = xi+1;
yi = yi;
pi = pi+4*xi+6;
}
else {
xi = xi+1;
yi = yi-1;
pi = pi+4*(xi-yi)+10;
}
}
}
```