```
#include<iostream>
#include<GL/glut.h>
                std;
#include<stdio.h>
#include<GL/glut.h>
#include<stdlib.h>
#include<math.h>
int xc=300,yc=300;

void displayPoint(int x, int y)
{
   glColor3f(0, 1, 0);
   glPointSize(2);
   glBegin(GL_POINTS);
   glVertex2i(x, y);
   glEnd();
}
void SimpleLine(float x1, float y1, float x2, float y2)
{
   float step;

   float dx = x2 - x1;
   float dy = y2 - y1;

      (abs(dx) >= abs(dy))
   {
      step = abs(dx);
   }

      step = abs(dy);

   float Xinc = dx / (float)step;
   float Yinc = dy / (float)step;
   float x = x1;
   float y = y1;

       (int i = 0; i <= step; i++)
   {
      displayPoint(x, y);
      x = x + Xinc;
      y = y + Yinc;
   }
   glFlush();
}
void plotPoint(int xc,int yc,int x,int y)
{
    glColor3f(0, 1, 1);
    glPointSize(2);
    glBegin(GL_POINTS);
        glVertex2i(xc+x,yc+y);
        glVertex2i(xc+y,yc+x);
        glVertex2i(xc-x,yc+y);
        glVertex2i(xc-y,yc+x);
        glVertex2i(xc-x,yc-y);
        glVertex2i(xc-y,yc-x);
        glVertex2i(xc+x,yc-y);
        glVertex2i(xc+y,yc-x);
    glEnd();
}
void midpoint_circle(int xc,int yc,int radius)
{
    int x=0;
    int y=radius;
    plotPoint(xc,yc,x,y);
    int p=1-radius;
```

```
    {
          (p<0)
        {
            x++;
            p=p+2*x+1;
        }
          (p>=0)
        {
            x++;
            y--;
            p=p+2*(x-y)+1;
        }
        plotPoint(xc,yc,x,y);
    }     (x<=y);
    glFlush();
}
void keyboard(unsigned char key, int x, int y)
{
int r=70;
          (key)
   {


        's':
   glClearColor(1.0, 1.0, 1.0, 1.0);
   glClear(GL_COLOR_BUFFER_BIT);
    midpoint_circle(xc,yc,100);
           ;

        'c':
      glClearColor(1.0, 1.0, 1.0, 1.0);
   glClear(GL_COLOR_BUFFER_BIT);
      midpoint_circle(xc,yc,50);
      midpoint_circle(xc,yc,200);
           ;

        'f':
    glClearColor(1.0, 1.0, 1.0, 1.0);
   glClear(GL_COLOR_BUFFER_BIT);

   midpoint_circle(xc,yc,r);
      midpoint_circle(xc+r,yc,r);
        midpoint_circle(xc-r,yc,r);
        midpoint_circle(xc,yc-r,r);
        midpoint_circle(xc,yc+r,r);

           ;
        'p':
     glClearColor(1.0, 1.0, 1.0, 1.0);
   glClear(GL_COLOR_BUFFER_BIT);

   midpoint_circle(xc,yc,r);
   midpoint_circle(xc+2*r,yc,r);
   midpoint_circle(xc,yc+2*r,r);
   midpoint_circle(xc-2*r,yc,r);
   midpoint_circle(xc,yc-2*r,r);
           ;
           'q':
     glClearColor(1.0, 1.0, 1.0, 1.0);
   glClear(GL_COLOR_BUFFER_BIT);

   midpoint_circle(xc,yc,r);

   midpoint_circle(xc+r+(r/1.5),yc,r/1.5);
   midpoint_circle(xc,yc+r+(r/1.5),r/1.5);
```

```
   midpoint_circle(xc-r-(r/1.5),yc,r/1.5);
   midpoint_circle(xc,yc-r-(r/1.5),r/1.5);
   midpoint_circle(xc+r+(r/5.2),yc+r+(r/5.2),r/1.5);
   midpoint_circle(xc-r-(r/5.2),yc-r-(r/5.2),r/1.5);
   midpoint_circle(xc+r+(r/5.2),yc-r-(r/5.2),r/1.5);
   midpoint_circle(xc-r-(r/5.2),yc+r+(r/5.2),r/1.5);

   }
   glutPostRedisplay();
}

void initialize(void)
{
   glClearColor(1.0, 1.0, 1.0, 1.0);
   glClear(GL_COLOR_BUFFER_BIT);
   // gluOrtho2D(l,r,b,t)
   gluOrtho2D(0, 600, 600, 0);
}

void primitives(void)
{

   glColor3f(1, 0, 0);
   SimpleLine(0, 300, 600, 300);
   SimpleLine(300, 0, 300, 600);
   glutKeyboardFunc(keyboard);
}
int main(int argc,char **argv)
{

   glutInit(&argc, argv);
   glutInitDisplayMode(GLUT_SINGLE);
   glutInitWindowPosition(0, 0);
   glutInitWindowSize(600, 600);
   glutCreateWindow("OpenGL - Mid Point Circle Drawing Algo - SIA03");
   initialize();


   glutDisplayFunc(primitives);
   cout<<"\n s:Simple Circle";
        cout<<"\n c:2 Concentric Circle";
        cout<<"\n f: intersecting circle";
        cout<<"\n p: one center circle and four circle around";
        cout<<"\n q: one center circle and eight circle around";
   glutMainLoop();
          0;
}
```