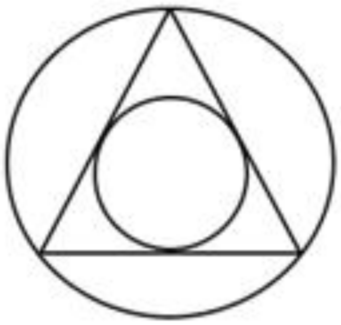
```
/
*===========================================================================
==========
Assignment No: 2
Batch       : H10
Roll No     : 23257
Date        :
Aim         : To implement line and circle drawing algorithms

Problem Statement:

Draw inscribed and Circumscribed circles in the triangle as shown as an example below (Use
any Circle drawing and Line drawing algorithms)
```

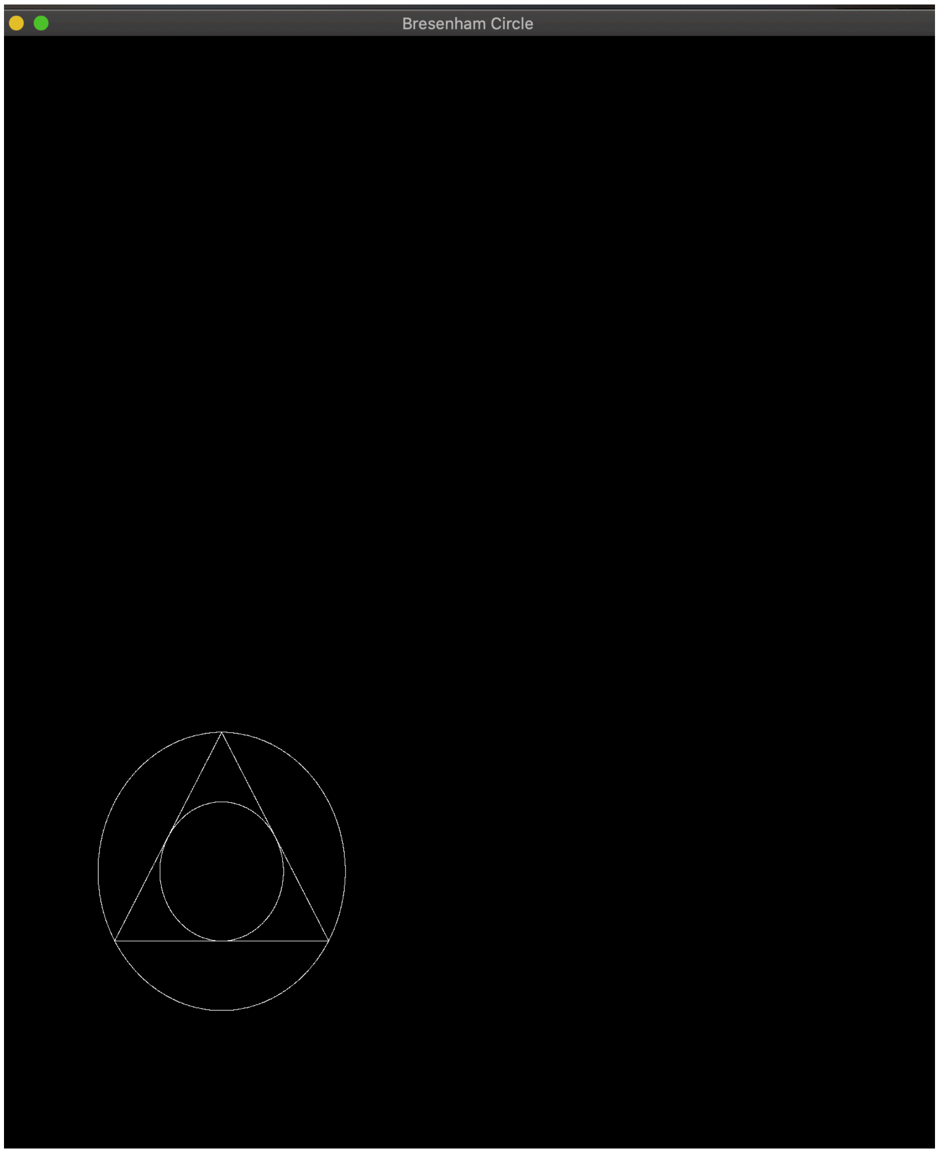


```
===========================================================================
============*/


#include <GLUT/glut.h>
#include <OpenGL/gl.h>
#include<iostream>
using namespace std;

int r,X,Y;

int abs(int x) {

	if(x<0)
		return (x*(-1));
	else
		return x;
}

int sign(int x) {

	if(x==0)
		return 0;

	else if(x<0)
		return -1;
```

```
        else return 1;
}

void SWAP(int *x,int *y) {

        int temp=*x;
        *x=*y;
        *y=temp;
}

void plot(int x,int y) {

        glBegin(GL_POINTS);
   glVertex2i(X+x,Y+y);
  glEnd();
}

void Bresenham(int x1,int y1,int x2,int y2) {

        int dx,dy,steps=1,x,y,i,P,sw=0,s1,s2;

        x=x1;
        y=y1;

        dx=abs(x2-x1);
        dy=abs(y2-y1);

        s1=sign(x2-x1);
        s2=sign(y2-y1);

        if(dy>dx) {

                SWAP(&dx,&dy);
                sw=1;
        }

        P=(2*dy)-dx;

        glBegin(GL_POINTS);
                glVertex2i(x,y);
        glEnd();

        while(steps<=dx) {

                if(P>=0) {

                        x=x+s1;
                        y=y+s2;
                        P=P+2*(dy-dx);
                }

                else {
                        if(sw==1) y=y+s2;
                        else      x=x+s1;

                        P=P+2*dy;
                }
                glBegin(GL_POINTS);
        glVertex2i(x,y);
    glEnd();
    steps++;
  }
}

void MidPointCircle(int r) {

        float p;
        int x,y;

        x=0;
        y=r;

  plot(x,y);

        p=(5/4)-r;

        while(y>x) {

                if(p<0) {

                  x=x+1;
                  p=p+(2*x)+3;
                }

                else if(p>=0) {

                  x=x+1;
                  y=y-1;
                  p=p+(2*(x-y))+5;
                }

    plot(x,y);
    plot(-x,y);
    plot(x,-y);
    plot(-x,-y);
    plot(y,x);
    plot(-y,x);
    plot(y,-x);
    plot(-y,-x);
  }
}

void BresenhamCircle(int r) {

        int p;
        int x,y;
```

```
            x=0;
            y=r;

  plot(x,y);

            p=3-(2*r);

            while(y>x) {

                        if(p<0) {

                          x=x+1;
                          p=p+(4*x)+6;
                        }

                        else if(p>=0) {

                          x=x+1;
                          y=y-1;
                          p=p+(4*(x-y))+10;
                        }

    plot(x,y);
    plot(-x,y);
    plot(x,-y);
    plot(-x,-y);
    plot(y,x);
    plot(-y,x);
    plot(y,-x);
    plot(-y,-x);
  }
}

void GenerateTriangle() {

  int x1,y1,x2,y2;

  x1=X-(1.732*r);
  y1=Y-r;
  x2=X;
  y2=Y+2*r;

  Bresenham(x1,y1,x2,y2);

  x1=X;
  y1=Y+2*r;
  x2=X+1.732*r;
  y2=Y-r;

  Bresenham(x1,y1,x2,y2);

  x1=X-(1.732*r);
  y1=Y-r;
  x2=X+1.732*r;
  y2=Y-r;

  Bresenham(x1,y1,x2,y2);
}


void DisplayMPC() {

  MidPointCircle(r);
  MidPointCircle(2*r);
  GenerateTriangle();

  glFlush();
}

void DisplayBCG() {

  BresenhamCircle(r);
  BresenhamCircle(2*r);
  GenerateTriangle();

  glFlush();
}

void init() {

          glClearColor(0.0, 0.0, 0.0, 0.0);
  glClear(GL_COLOR_BUFFER_BIT);
  glColor3f(1.0, 1.0, 1.0);

  glMatrixMode(GL_PROJECTION);
          glLoadIdentity();
  gluOrtho2D(-1000,1000,-1000,1000);
}

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

          int ch,flag=0;

          do {

            cout<<"===================================\n";
            cout<<"Select Correct Option\n";
            cout<<"===================================\n";
            cout<<"1. Enter Coordinates\n";
            cout<<"2. Generate Circle by DDA\n";
            cout<<"3. Generate Circle by Bresenham\n";
            cout<<"4. Exit\n";
            cout<<"Your Chice: ";
            cin>>ch;
            cout<<"===================================\n";

    if(ch==1 || flag==1 || ch==4) {
            switch(ch) {

              case 1: {

                cout<<"Centre Coordinates:\n";
                cout<<"X: ";
                cin>>X;

                cout<<"Y: ";
                cin>>Y;

                cout<<"Radius: ";
                cin>>r;

                flag=1;
                break;
              }

              case 2: {

                    glutInit(&argc, argv);
                  glutInitDisplayMode(GLUT_SINGLE);
                  glutInitWindowSize(800,900);
        glutInitWindowPosition(320,240);
        glutCreateWindow("MidPoint Circle");
        init();
        glutDisplayFunc(DisplayMPC);
        glutMainLoop();
                  break;
              }

              case 3: {

                    glutInit(&argc, argv);
                  glutInitDisplayMode(GLUT_SINGLE);
                  glutInitWindowSize(800,900);
        glutInitWindowPosition(320,240);
        glutCreateWindow("Bresenham Circle");
        init();
        glutDisplayFunc(DisplayBCG);
        glutMainLoop();
                  break;
              }

              case 4: cout<<"End\n";
                     break;

             default: cout<<"Select correct Option\n\n";
            }
}
          else cout<<"Please Accept Coordinates First\n\n";
        }
        while(ch!=4);
```

```
  return 0;
}

/*

Output:

Someshwars-MacBook-Pro:Computer Graphics someshwargaikwad$ ./Assignment2
====================================
Select Correct Option
====================================
1. Enter Coordinates
2. Generate Circle by DDA
3. Generate Circle by Bresenham
4. Exit
Your Chice: 1
====================================
Centre Coordinates:
X: 0
Y: 0
Radius: 250
====================================
Select Correct Option
====================================
1. Enter Coordinates
2. Generate Circle by DDA
3. Generate Circle by Bresenham
4. Exit
Your Chice: 3
====================================
```


Bresenham Circle

```
*/
```