**Shudhanshu Shrotriya**

**500086221**

**Batch-4**

**Experiment - 7**

**Computer Graphics Lab**

**Mid-Point Circle Drawing Algorithm**

**Algorithm:**

**Step1:** Put x =0, y =r in equation 2  
            We have p=1-r

**Step2:** Repeat steps while x ≤ y  
            Plot (x, y)  
            If (p<0)  
Then set p = p + 2x + 3  
Else  
            p = p + 2(x-y)+5  
            y =y - 1 (end if)  
            x =x+1 (end loop)

**Step3:** End

**Code:**

#include <stdio.h>

#include <iostream>

#include <GL/glut.h>

using namespace std;

int pntX1, pntY1, r;

void plot(int x, int y)

{

glBegin(GL\_POINTS);

glVertex2i(x+pntX1, y+pntY1);

glEnd();

}

void myInit (void)

{

glClearColor(1.0, 1.0, 1.0, 0.0);

glColor3f(0.0f, 0.0f, 0.0f);

glPointSize(4.0);

glMatrixMode(GL\_PROJECTION);

glLoadIdentity();

gluOrtho2D(0.0, 640.0, 0.0, 480.0);

}

void midPointCircleAlgo()

{

int x = 0;

int y = r;

float decision = 5/4 - r;

plot(x, y);

while (y > x)

{

if (decision < 0)

{

x++;

decision += 2\*x+1;

}

else

{

y--;

x++;

decision += 2\*(x-y)+1;

}

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 myDisplay(void)

{

glClear (GL\_COLOR\_BUFFER\_BIT);

glColor3f (0.0, 0.0, 0.0);

glPointSize(1.0);

midPointCircleAlgo();

glFlush ();

}

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

{

cout << "Enter the coordinates of the center:\n\n" << endl;

cout << "X-coordinate : "; cin >> pntX1;

cout << "\nY-coordinate : "; cin >> pntY1;

cout << "\nEnter radius : "; cin >> r;

glutInit(&argc, argv);

glutInitDisplayMode (GLUT\_SINGLE | GLUT\_RGB);

glutInitWindowSize (640, 480);

glutInitWindowPosition (100, 150);

glutCreateWindow ("Mid Point Circle Drawing Algorithm");

glutDisplayFunc(myDisplay);

myInit ();

glutMainLoop();

}

**Output:**

