**Day – 5**

**CODE :-**

#include "circleellipsewidget.h"

#include <QPainter>

#include <QPaintEvent>

CircleEllipseWidget::CircleEllipseWidget(QWidget \*parent)

: QWidget(parent)

{

setFixedSize(400, 400); // Adjust as needed

}

void CircleEllipseWidget::paintEvent(QPaintEvent \*event)

{

QPainter painter(this);

painter.fillRect(rect(), Qt::black); // Black background

painter.setPen(Qt::white); // White pen

int cx = width() / 2;

int cy = height() / 2;

// Circle radius

int r = 100;

// Ellipse semi-axes

int a = 100;

int b = 50;

// Draw a circle (like the sphere outline)

drawMidpointCircle(painter, cx, cy, r);

drawMidpointCircle(painter, cx, cy, r/2);

drawMidpointCircle(painter, cx, cy, r/4);

drawMidpointCircle(painter, cx, cy, r/8);

// Draw an ellipse (horizontal cross-section)

drawMidpointEllipse(painter, cx, cy, a, b);

drawMidpointEllipse(painter, cx, cy, a/2, b/2);

drawMidpointEllipse(painter, cx, cy, a/4, b/4);

}

void CircleEllipseWidget::drawMidpointCircle(QPainter &painter, int cx, int cy, int r)

{

int x = 0, y = r;

int p = 1 - r; // Initial decision parameter

while (x <= y) {

// Plot the 8 symmetric points

painter.drawPoint(cx + x, cy + y);

painter.drawPoint(cx - x, cy + y);

painter.drawPoint(cx + x, cy - y);

painter.drawPoint(cx - x, cy - y);

painter.drawPoint(cx + y, cy + x);

painter.drawPoint(cx - y, cy + x);

painter.drawPoint(cx + y, cy - x);

painter.drawPoint(cx - y, cy - x);

if (p < 0) {

p += 2 \* x + 1;

} else {

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

y--;

}

x++;

}

}

void CircleEllipseWidget::drawMidpointEllipse(QPainter &painter, int cx, int cy, int a, int b)

{

int x = 0, y = b;

int a2 = a \* a, b2 = b \* b;

// Initial decision parameter for region 1

int err = b2 - a2 \* b + a2 / 4;

auto plotPoints = [&](int px, int py) {

painter.drawPoint(cx + px, cy + py);

painter.drawPoint(cx - px, cy + py);

painter.drawPoint(cx + px, cy - py);

painter.drawPoint(cx - px, cy - py);

};

plotPoints(x, y);

// Region 1

while (b2 \* x < a2 \* y) {

x++;

if (err < 0) {

err += b2 \* (2 \* x + 1);

} else {

y--;

err += b2 \* (2 \* x + 1) - 2 \* a2 \* y;

}

plotPoints(x, y);

}

// Region 2

err = b2 \* (x + 1)\*(x + 1) + a2 \* (y - 1)\*(y - 1) - a2 \* b2;

while (y >= 0) {

y--;

if (err > 0) {

err -= a2 \* (2 \* y + 1);

} else {

x++;

err += b2 \* (2 \* x + 1) - 2 \* a2 \* y;

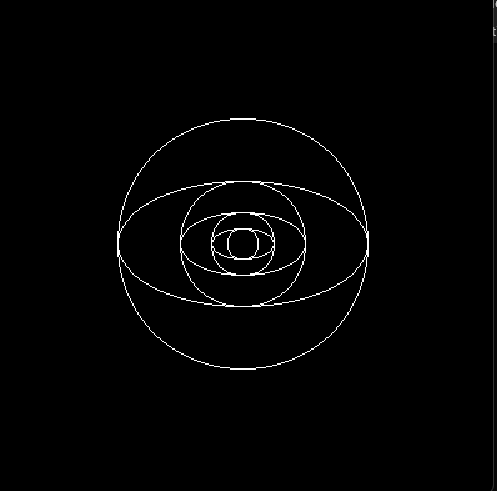
}

plotPoints(x, y);

}

}

**OUTPUT :-**

****