#include <graphics.h>

#include <iostream>

using namespace std;

class MidpointCircle {

public:

void midPointCircleDraw(int x\_centre, int y\_centre, int r, int color) {

int x = r, y = 0, P;

setcolor(color); // Set the current drawing color

putpixel(x + x\_centre, y + y\_centre, color); // Draw the initial point

if (r > 0) {

putpixel(x + x\_centre, -y + y\_centre, color); // Mirror points

putpixel(y + x\_centre, x + y\_centre, color);

putpixel(-y + x\_centre, x + y\_centre, color);

}

P = 1 - r;

while (x > y) {

y++;

if (P <= 0) {

P = P + 2 \* y + 1;

} else {

x--;

P = P + 2 \* y - 2 \* x + 1;

}

if (x < y) break; // Break if the circle is completed

// Draw all the symmetrical points

putpixel(x + x\_centre, y + y\_centre, color);

putpixel(-x + x\_centre, y + y\_centre, color);

putpixel(x + x\_centre, -y + y\_centre, color);

putpixel(-x + x\_centre, -y + y\_centre, color);

delay(50); // Control speed of drawing

if (x != y) {

putpixel(y + x\_centre, x + y\_centre, color);

putpixel(-y + x\_centre, x + y\_centre, color);

putpixel(y + x\_centre, -x + y\_centre, color);

putpixel(-y + x\_centre, -x + y\_centre, color);

delay(50);

}

}

}

};

void drawRainbow(int x\_centre, int y\_centre, int start\_radius) {

MidpointCircle ob;

int colors[] = {4, 14, 7, 2, 1, 9, 13}; // Red, Orange, Yellow, Green, Blue, Indigo, Violet

int radius = start\_radius;

for (int i = 0; i < 7; i++) {

ob.midPointCircleDraw(x\_centre, y\_centre, radius, colors[i]); // Draw each band with different color

radius -= 10; // Decrease radius for next circle (make bands smaller)

}

}

int main() {

int gd = DETECT, gm;

initgraph(&gd, &gm, NULL); // Initialize the graphics mode

int x, y, radius;

cout << "Enter the center coordinates (x, y): ";

cin >> x >> y;

cout << "Enter the radius of the outermost rainbow circle: ";

cin >> radius;

drawRainbow(x, y, radius); // Call the function to draw the rainbow

getch(); // Wait for user input to close the graphics window

closegraph(); // Close the graphics window

return 0;

}