'''

Name:- Riya Manoj Wagh

Class:- SE - Computer-B (SB3)

Roll No: - 65

Subject:- Computer Graphics

ii) Write a C++ program to implement Midpoint

circle drawing algorithm to draw a dotted/dashed

circle. Apply the concept of encapsulation.

'''

#include <iostream>

#include <graphics.h>

#include <cmath>

using namespace std;

class MidpointCircle {

private:

int xc, yc, r;

int interval;

// Function to draw a pixel (dotted or dashed)

void drawPixel(int x, int y) {

if ((x + y) % interval == 0) {

putpixel(xc + x, yc + y, WHITE);

putpixel(xc - x, yc + y, WHITE);

putpixel(xc + x, yc - y, WHITE);

putpixel(xc - x, yc - y, WHITE);

putpixel(xc + y, yc + x, WHITE);

putpixel(xc - y, yc + x, WHITE);

putpixel(xc + y, yc - x, WHITE);

putpixel(xc - y, yc - x, WHITE);

}

}

public:

// Constructor to initialize circle properties

MidpointCircle(int centerX, int centerY, int radius, int intervalSpacing = 5) {

xc = centerX;

yc = centerY;

r = radius;

interval = intervalSpacing; // Defines the spacing between the

dotted/dashed pixels

}

// Midpoint Circle Drawing Algorithm to draw the dotted circle

void drawCircle() {

int x = 0;

int y = r;

int p = 1 - r;

// Draw initial points for the first octant of the circle

drawPixel(x, y);

while (x < y) {

x++;

if (p < 0)

p = p + 2 \* x + 1;

else {

y--;

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

}

// Draw all eight octants

drawPixel(x, y);

}

}

};

int main() {

int gd = DETECT, gm;

initgraph(&gd, &gm, NULL);

int centerX, centerY, radius, intervalSpacing;

// Taking inputs for circle properties

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

cin >> centerX >> centerY;

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

cin >> radius;

cout << "Enter the interval spacing for dotted/dashed effect (recommended

value: 5): ";

cin >> intervalSpacing;

// Create an object of MidpointCircle and draw the circle

MidpointCircle circle(centerX, centerY, radius, intervalSpacing);

circle.drawCircle();

// Wait for user input to close the graphics window

getch();

closegraph();

return 0;

}