#include <iostream>

#include <graphics.h>

#include <math.h>

using namespace std;

void drawCircle(int xc, int yc, int x, int y)

{

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);

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

}

void Bresenhams(int xc, int yc, int r)

{

int x = 0;

int y = r;

int d = 3 - 2 \* r;

while (y >= x)

{

x++;

if (d < 0)

{

d = d + 4 \* x + 6;

}

else

{

y--;

d = d + 4 \* (x - y) + 10;

}

drawCircle(xc, yc, x, y);

delay(50);

}

}

void dda(int x1, int y1, int x2, int y2)

{

int dx = x2 - x1;

int dy = y2 - y1;

int steps = dx > dy ? dx : dy;

float xInc = dx / (float)steps;

float yInc = dy / (float)steps;

float x = x1;

float y = y1;

for (int i = 0; i <= steps; i++)

{

putpixel(x, y, 14);

x += xInc;

y += yInc;

}

}

void pattern(int xmin, int ymin, int xmax, int ymax)

{ // Square

dda(xmin, ymin, xmax, ymin);

dda(xmax, ymin, xmax, ymax);

dda(xmin, ymax, xmax, ymax);

dda(xmin, xmin, xmin, xmax);

float xmid = (xmin + xmax) / 2;

float ymid = (ymin + ymax) / 2;

// Diamond inside square

dda(xmin, ymid, xmid, ymax);

dda(xmid, ymax, xmax, ymid);

dda(xmid, ymin, xmax, ymid);

dda(xmid, ymin, xmin, ymid);

// circle

float side = sqrt(pow((ymin - ymid), 2) + pow((xmid - xmin), 2));

float r = side / 2;

Bresenhams(xmid, ymid, r);

}

int main()

{

int gd = DETECT, gm;

initgraph(&gd, &gm, NULL);

pattern(200, 200, 400, 400);

getch();

closegraph();

return 0;

}