**Lab report - 02**

*Course title: Computer Graphics Laboratory*

*Course code: CSE-304*

*3rd Year 1st Semester 2022*

**Date of Submission**: 04/06/2023



**Submitted to-**

Dr. Mohammad Shorif Uddin (Professor)

Dr. Morium Akter (Associate Professor)

*Department of Computer Science and Engineering*

*Jahangirnagar University*

*Savar, Dhaka-1342*

|  |  |  |
| --- | --- | --- |
| **Name** | **Class roll** | **Exam roll** |
| Hasneen Tamanna Totinee | 362 | 202174 |

**1. Midpoint Circle algorithm**

Code:

#include <bits/stdc++.h>

#include <graphics.h>

using namespace std;

// Function to draw a circle using Midpoint Circle Algorithm

void drawCircle(int xc, int yc, int radius)

{

int x = 0;

int y = radius;

int p = 1 - radius;

// Set initial point

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

while (x < y)

{

x++;

if (p < 0)

{

p += 2 \* x + 1;

}

else

{

y--;

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

}

// Plotting points in all octants

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);

}

}

int main()

{

int gd = DETECT, gm;

initgraph(&gd, &gm, "");

int xc = 250; // X-coordinate of the center of the circle

int yc = 250; // Y-coordinate of the center of the circle

int radius = 100; // Radius of the circle

drawCircle(xc, yc, radius);

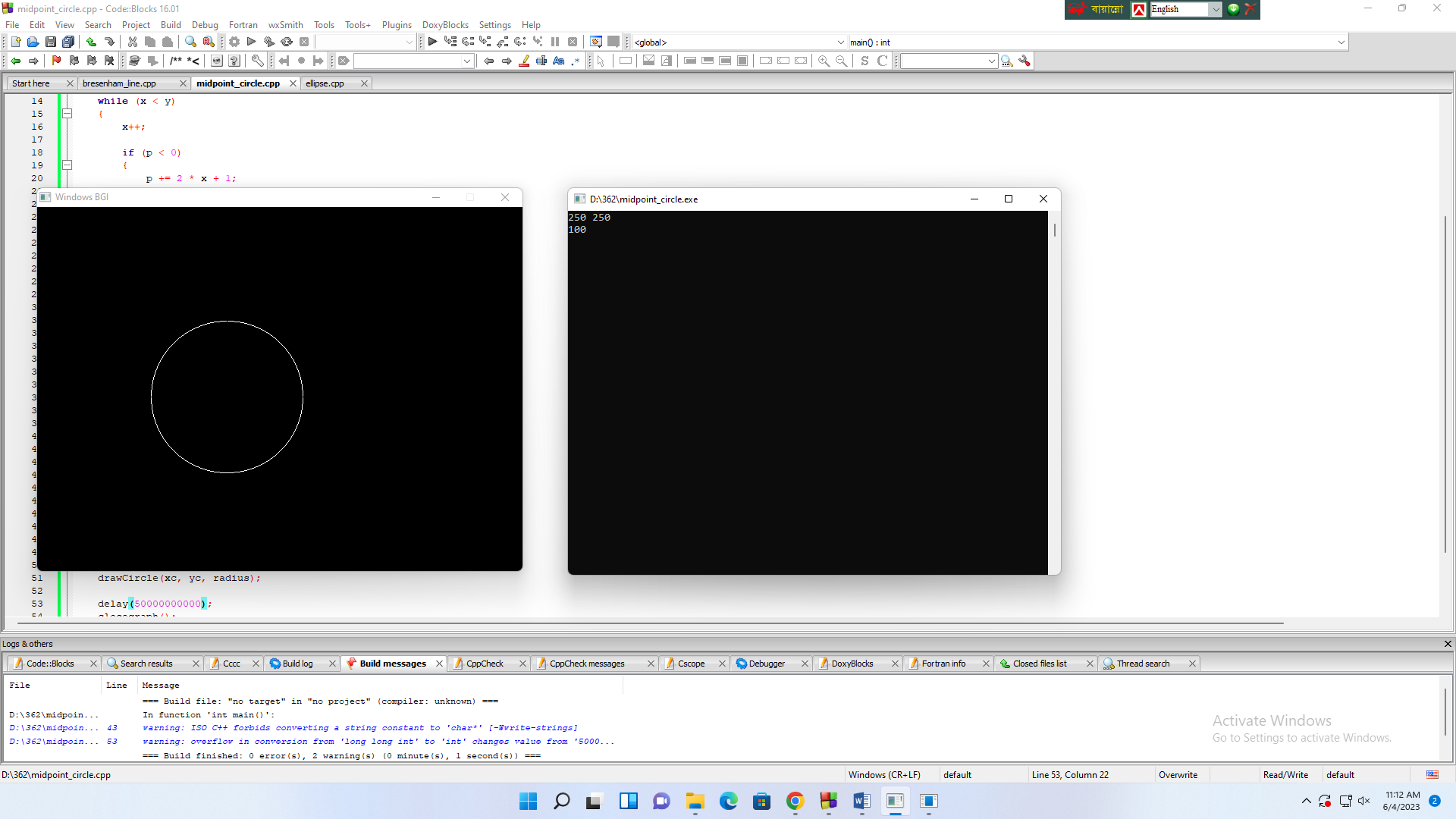
delay(5000);

closegraph();

return 0;

}

Output:



**2. Scan conversion of an Ellipse**

Code:

#include <bits/stdc++.h>

#include <graphics.h>

using namespace std;

// Function to plot points in all quadrants

void plotEllipsePoints(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 - x, yc - y, WHITE);

}

// Function to draw ellipse using midpoint algorithm

void drawEllipse(int xc, int yc, int rx, int ry)

{

int x = 0;

int y = ry;

// Decision parameters

int rxSq = rx \* rx;

int rySq = ry \* ry;

int twoRxSq = 2 \* rxSq;

int twoRySq = 2 \* rySq;

int p;

int px = 0;

int py = twoRxSq \* y;

// Plot initial point in all quadrants

plotEllipsePoints(xc, yc, x, y);

// Region 1

p = rySq - (rxSq \* ry) + (0.25 \* rxSq);

while (px < py)

{

x++;

px += twoRySq;

if (p < 0)

{

p += rySq + px;

}

else

{

y--;

py -= twoRxSq;

p += rySq + px - py;

}

plotEllipsePoints(xc, yc, x, y) }

// Region 2

p = rySq \* (x + 0.5) \* (x + 0.5) + rxSq \* (y - 1) \* (y - 1) - rxSq \* rySq;

while (y > 0)

{

y--;

py -= twoRxSq;

if (p > 0)

{

p += rxSq - py;

}

else

{

x++;

px += twoRySq;

p += rxSq - py + px;

}

plotEllipsePoints(xc, yc, x, y);

}

}

int main()

{

int gd = DETECT, gm;

initgraph(&gd, &gm, "");

int xc,yc,rx,ry;

cin>> xc>> yc>> rx>> ry;

// X-coordinate of the center of the ellipse

// Y-coordinate of the center of the ellipse

// X-radius of the ellipse

// Y-radius of the ellipse

drawEllipse(xc, yc, rx, ry);

delay(5000000000);

closegraph();

return 0;

}

Output:

