

**Circle using mid point::**

**Code:**

#include <iostream>

#include <graphics.h>

using namespace std;

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

int x = 0;

int y = radius;

int p = 1 - radius;

while (x <= y) {

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

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

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

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

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

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

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

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

x++;

if (p < 0) {

p += 2 \* x + 1;

} else {

y--;

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

}

}

}

int main() {

int xc, yc, radius;

cout << "Enter center coordinates (xc and yc): ";

cin >> xc >> yc;

cout << "Enter radius: ";

cin >> radius;

int gd = DETECT, gm;

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

drawCircle(xc, yc, radius);

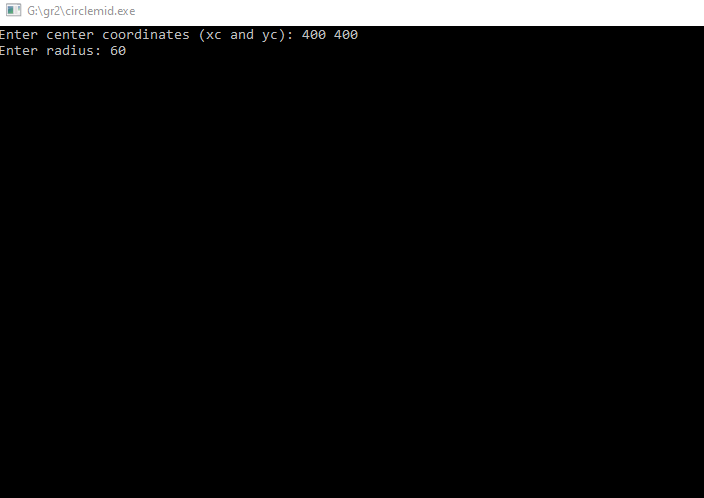
getch();

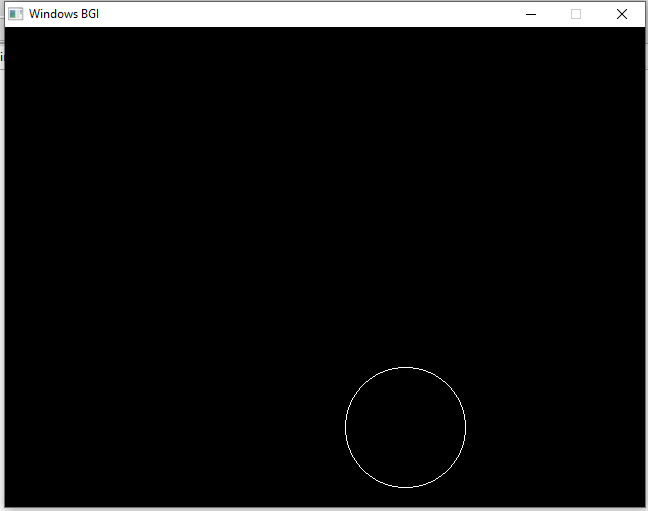
closegraph();

return 0;

}

Output:





Ellipse:

Code:

#include <graphics.h>

int main()

{

int gd = DETECT, gm;

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

ellipse(300, 200, 0, 360, 100, 50); // Draw ellipse

getch();

closegraph(); // Close graphics mode

return 0;

}



Code 2:

#include <iostream>

#include <graphics.h>

using namespace std;

void drawEllipse(int a, int b) {

int x = 0; // x coordinate

int y = b; // y coordinate

int d1 = (b \* b) - (a \* a \* b) + (a \* a \* 0.25);

int dx = 2 \* b \* b \* x;

int dy = 2 \* a \* a \* y;

while (dx < dy) {

putpixel(x, y, WHITE);

if (d1 < 0) {

x++;

dx += 2 \* b \* b;

d1 += dx + b \* b;

}

else {

x++;

y--;

dx += 2 \* b \* b;

dy -= 2 \* a \* a;

d1 += dx - dy + b \* b;

}

}

int d2 = ((b \* b) \* ((x + 0.5) \* (x + 0.5))) +

((a \* a) \* ((y - 1) \* (y - 1))) -

(a \* a \* b \* b);

while (y >= 0) {

putpixel(x, y, WHITE);

if (d2 > 0) {

y--;

dy -= 2 \* a \* a;

d2 += a \* a - dy;

}

else {

y--;

x++;

dx += 2 \* b \* b;

dy -= 2 \* a \* a;

d2 += dx - dy + a \* a;

}

}

}

int main() {

int a, b;

cout << "Enter major and minor axis lengths (a and b): ";

cin >> a >> b;

int gd = DETECT, gm;

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

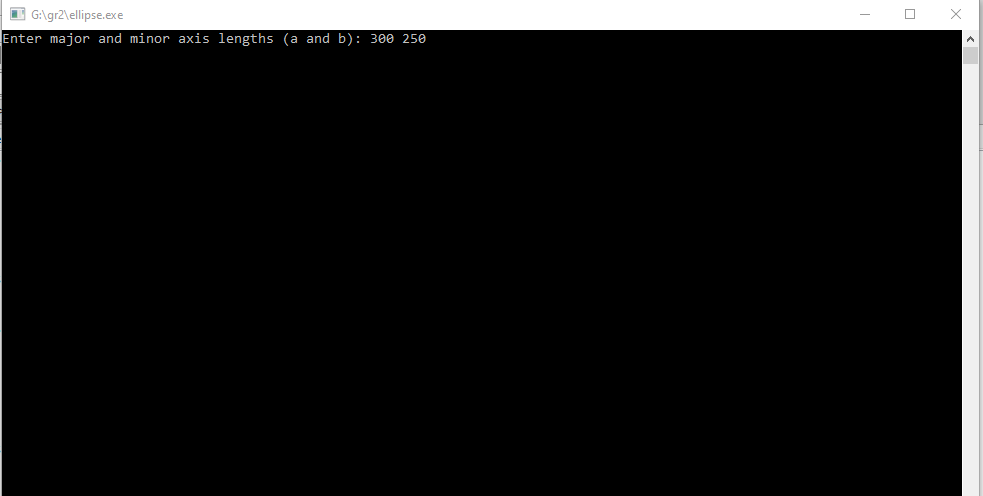
drawEllipse(a, b);

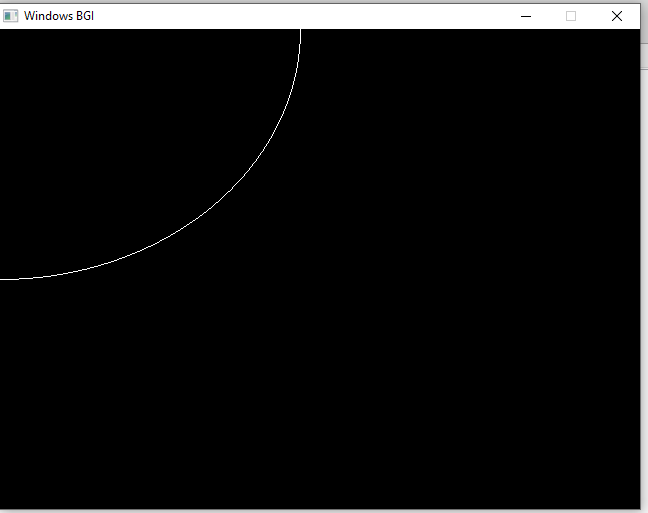
getch();

closegraph();

return 0;

}





Ellipse ::

Code 3:

#include <iostream>

#include <graphics.h>

// 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 = 250; // X-coordinate of the center of the ellipse

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

int rx = 150; // X-radius of the ellipse

int ry = 100; // Y-radius of the ellipse

drawEllipse(xc, yc, rx, ry);

// delay(5000);

getch();

closegraph();

return 0;

}

