**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***

***And***

***Dr. Morium Akter***

*Department of Computer Science and Engineering*

*Jahangirnagar University*

*Savar, Dhaka-1342*

|  |  |  |  |
| --- | --- | --- | --- |
| **Sl** | Class Roll | Exam Roll | Name |
| 01 | 355 | 202167 | Nuzhat Nairy Afrin |

**1.Scan Conversion of an Ellipse using Mid Point Algorithm**

**Source Code:**

#include<bits/stdc++.h>

#include<iostream>

#include<graphics.h>

using namespace std;

void drawellipse(int xc, int yc, int a, int b)

{

int x=0;

int y=b;

int a\_sqr = a\*a;

int b\_sqr = b\*b;

int two\_a\_sqr = 2\*a\_sqr;

int two\_b\_sqr = 2\*b\_sqr;

int p;

int px = 0;

int py = two\_a\_sqr\*y;

p = b\_sqr - (a\_sqr\*b)+(0.25\*a\_sqr);

while(px<py)

{

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

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

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

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

x++;

px += two\_b\_sqr;

if(p<0)

{

p += b\_sqr+px;

}

else{

y--;

py -= two\_a\_sqr;

p += b\_sqr + px-py;

}

}

while(y>=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);

y--;

py -= two\_a\_sqr;

if(p>0)

{

p += a\_sqr - py;

}

else

{

x++;

px += two\_a\_sqr;

p += a\_sqr - py+px;

}

}

}

int main()

{

int gd = DETECT, gm;

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

int xc,yc,a,b;

cout << "Enter the center coordinate of the ellipse: " << endl;

cin >> xc >> yc;

cout << "Enter the length of the major axis: " << endl;

cin >> a;

cout <<"Enter the length of the minor axis: "<< endl;

cin >> b;

drawellipse(xc,yc,a,b);

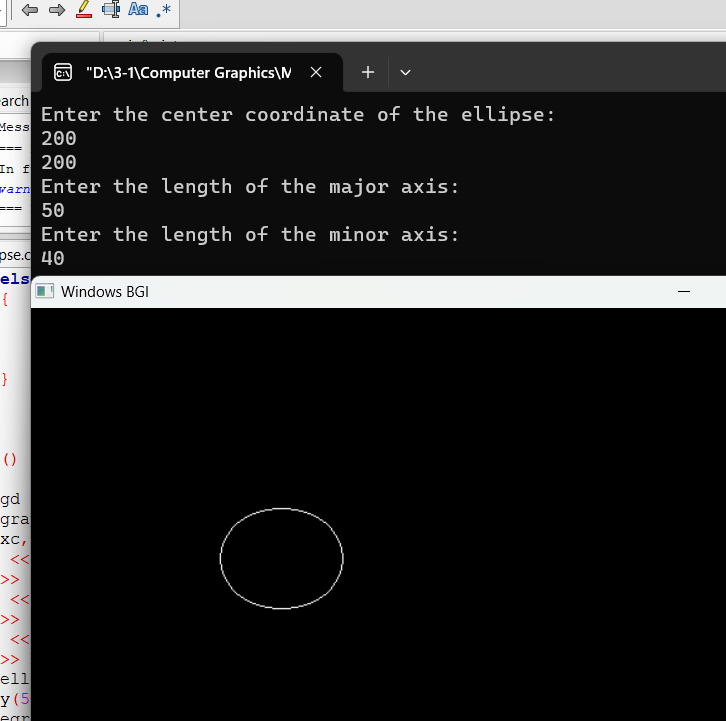
delay(500000);

closegraph();

return 0;

}

**Output:**

****

**2.Scan Conversion of a Circle Using Mid Point algorithm**

**Source Code:**

#include <iostream>

#include <cmath>

#include <graphics.h>

using namespace std;

void drawCircleMidpoint(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 - 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);

if (p < 0) {

p += 2\*x +3;

} else {

p += 2 \* (x - y) + 5;

y--;

}

x++;

}

}

int main() {

int gd = DETECT, gm;

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

int xc, yc, radius;

cout << "Enter the coordinates of the center (xc, yc): ";

cin >> xc >> yc;

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

cin >> radius;

drawCircleMidpoint(xc, yc, radius);

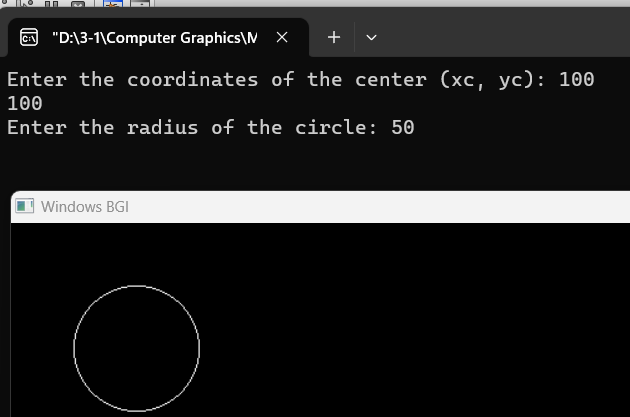
delay(5000);

closegraph();

return 0;

}

**Output:**

****