**Lab Report:2**

**Title:**

*Course title: Computer Graphics*

*Course code: CSE-304*

*3rd year 1st Semester*

**Date of Submission**: 04.06.2023

****

**Submitted to-**

Dr. Md. Shorif Uddin

***Professor***

Dr. Morium Akter

***Assistant Professor***

*Department of Computer Science and Engineering*

*Jahangirnagar University*

*Savar, Dhaka-1342*

|  |  |  |  |
| --- | --- | --- | --- |
| **Sl** | Class Roll | Exam Roll | Name |
| 01 | 2147 | 191323 | Shabrina Akter Shahana |

**1.Source Code of Scan Conversion of Ellipse using Midpoint Algorithm:**

#include<bits/stdc++.h>

#include <iostream>

#include <graphics.h>

#include <conio.h>

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);

}

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

{

int rx2 = rx \* rx;

int ry2 = ry \* ry;

int twoRx2 = 2 \* rx2;

int twoRy2 = 2 \* ry2;

int p;

int x = 0;

int y = ry;

int px = 0;

int py = twoRx2 \* y;

plotEllipsePoints(xc, yc, x, y);

p = round(ry2 - (rx2 \* ry) + (0.25 \* rx2));

while (px < py){

x++;

px += twoRy2;

if (p < 0) {

p += ry2 + px;}

else{

y--;

py -= twoRx2;

p += ry2 + px - py;}

plotEllipsePoints(xc, yc, x, y);

}

p = round(ry2 \* (x + 0.5) \* (x + 0.5) + rx2 \* (y - 1) \* (y - 1) - rx2 \* ry2);

while (y > 0){

y--;

py -= twoRx2;

if (p > 0){

p += rx2 - py;}

else {

x++;

px += twoRy2;

p += rx2 - py + px;}

plotEllipsePoints(xc, yc, x, y); }}

int main(){

int gd = DETECT, gm;

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

int xc = 200; // x-coordinate of the ellipse center

int yc = 200; // y-coordinate of the ellipse center

int rx = 100; // x-radius of the ellipse

int ry = 50; // y-radius of the ellipse

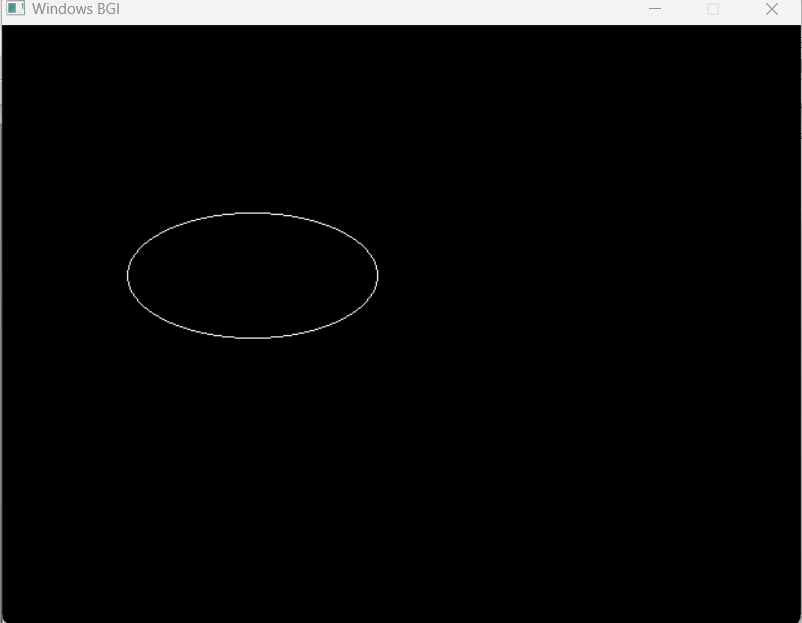
scanConvertEllipse(xc, yc, rx, ry);

getch();

closegraph();

return 0;}

**OUTPUT:**



**1.Source Code of Scan Conversion of Circle using Midpoint Algorithm:**

#include <iostream>

#include<bits/stdc++.h>

#include <graphics.h>

using namespace std;

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

int x = 0; // Initial x coordinate

int y = radius; // Initial y coordinate

int d = 1 - radius; // Initial decision parameter

int gd = DETECT, gm;

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

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);

while (x <= y) {

if (d < 0) {

x++;

d += 2 \* x + 1;}

else {

x++;

y--;

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

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

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

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

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

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

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

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

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

} delay(588000);

closegraph();}

int main() {

int radius; //100

int xc; // X-coordinate of the center 320

int yc; // Y-coordinate of the center 240

cout<<"Enter the x axis and y axis:"<<endl;

cin>>xc;

cin>>yc;

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

cin>>radius;

drawCircle(radius, xc, yc);

return 0;}

**OUTPUT:**

