**Lab Report: 01**

**Title: Scan conversion**

*Course title: Computer Graphics Laboratory*

*Course code: CSE-304*

*3rd Year 1st Semester Examination 2022*

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

****

**Submitted to-**

*Dr. Mohammad Shorif Uddin*

*Professor*

*Department of Computer Science and Engineering*

*Jahangirnagar University*

*&*

*Dr. Morium Akter*

*Associate Professor*

*Department of Computer Science and Engineering*

*Jahangirnagar University*

*Savar, Dhaka-1342*

|  |  |  |
| --- | --- | --- |
| Class Roll | Exam Roll | Name |
| 356 | 20200650726 | Syeda Faria Sithi |

**Experiment no: 01**

**Experiment Name: Scan conversion of a point**

Source Code:

#include<bits/stdc++.h>

#include <graphics.h>

#include <conio.h>

using namespace std;

int main()

{

int x, y;

cout << "\nEnter the x-coordinate: ";

cin >> x;

cout << "\nEnter the y-coordinate: ";

cin >> y;

int gd = DETECT, gm;

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

putpixel(x, y, WHITE);

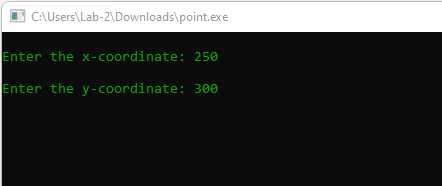
getch();

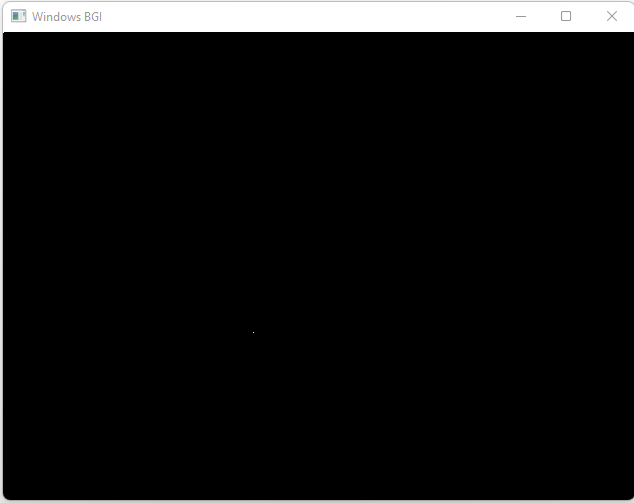
closegraph();

return 0;

}

**Output:**

****

****

**Experiment no: 02**

**Experiment Name: Scan conversion of a line using DDA Algorithm**

Source Code:

#include <graphics.h>

#include <iostream>

using namespace std;

int round(float n)

{

if (n - (int)n < 0.5)

return (int)n;

return (int)(n + 1);

}

void DDALine(int x0, int y0, int x1, int y1)

{

int gd = DETECT, gm;

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

int dx = x1 - x0;

int dy = y1 - y0;

int step;

if (abs(dx) > abs(dy))

step = abs(dx);

else

step = abs(dy);

float x\_incr = (float)dx / step;

float y\_incr = (float)dy / step;

float x = x0;

float y = y0;

cleardevice();

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

{

putpixel(round(x), round(y), WHITE);

delay(10);

x += x\_incr;

y += y\_incr;

}

getch();

closegraph();

}

int main()

{

int x1, y1, x2, y2;

cout << "\nEnter the starting x-coordinate: ";

cin >> x1;

cout << "Enter the starting y-coordinate: ";

cin >> y1;

cout << "\nEnter the ending x-coordinate: ";

cin >> x2;

cout << "Enter the ending y-coordinate: ";

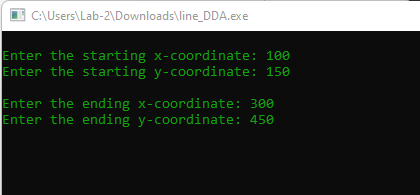
cin >> y2;

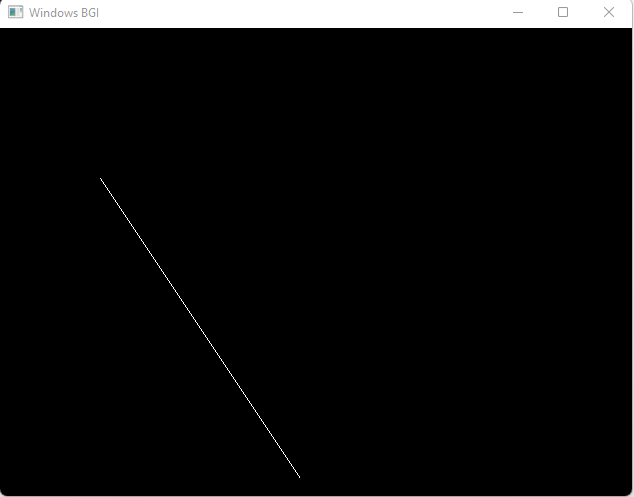
DDALine(x1, y1, x2, y2);

return 0;

}

**Output:**

****

****

**Experiment no: 03**

# Experiment Name: Scan conversion of a line using Bresenham's Line Algorithm

Source Code:

#include <graphics.h>

#include <iostream>

using namespace std;

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

{

int gd = DETECT, gm;

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

int m\_new = 2 \* (y2 - y1);

int slope\_error\_new = m\_new - (x2 - x1);

for (int x = x1, y = y1; x <= x2; x++) {

putpixel(x, y, WHITE);

delay(10);

slope\_error\_new += m\_new;

if (slope\_error\_new >= 0) {

y++;

slope\_error\_new -= 2 \* (x2 - x1);

}

}

getch();

closegraph();

}

int main()

{

int x1, y1, x2, y2;

cout << "\nEnter the starting x-coordinate: ";

cin >> x1;

cout << "Enter the starting y-coordinate: ";

cin >> y1;

cout << "\nEnter the ending x-coordinate: ";

cin >> x2;

cout << "Enter the ending y-coordinate: ";

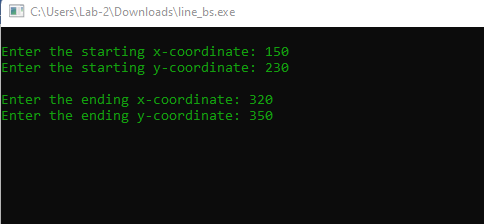
cin >> y2;

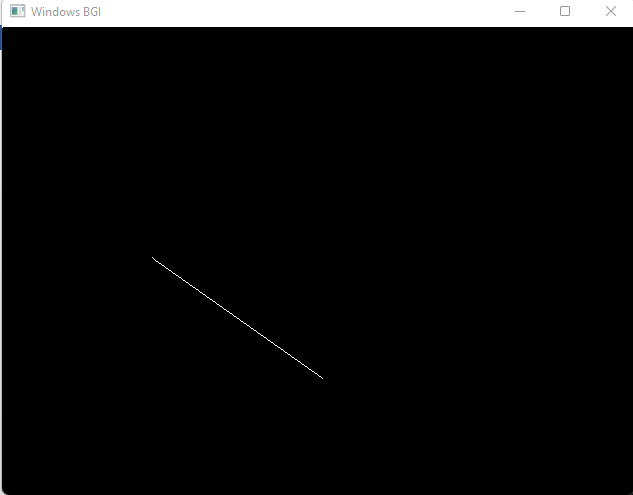
bresenham(x1, y1, x2, y2);

return 0;

}

**Output:**

****

****

**Experiment no: 04**

**Experiment Name: Scan conversion of a circle using Bresenham's circle Algorithm**

Source Code:

#include <dos.h>

#include<bits/stdc++.h>

#include <graphics.h>

#include <conio.h>

using namespace std;

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

{

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

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

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

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

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

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

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

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

}

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

{

int x = 0, y = r;

int d = 3 - 2 \* r;

drawCircle(xc, yc, x, y);

while (y >= x)

{

x++;

if (d > 0)

{

y--;

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

}

else

d = d + 4 \* x + 6;

drawCircle(xc, yc, x, y);

delay(50);

}

getch();

closegraph();

}

int main()

{

int x, y, r;

cout << "\nEnter the x-coordinate of the center: ";

cin >> x;

cout << "Enter the y-coordinate of the center: ";

cin >> y;

cout << "\nEnter the Radius of the circle: ";

cin >> r;

int gd = DETECT, gm;

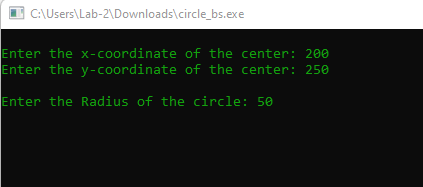
initgraph(&gd, &gm, "");

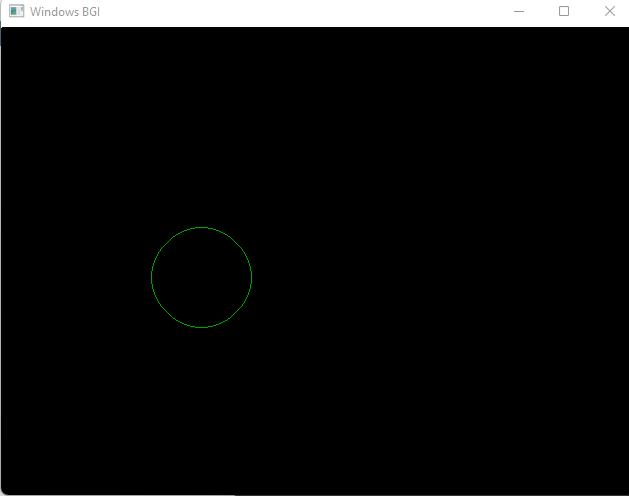
circleBres(x, y, r);

return 0;

}

**Output:**

****

****