**Lab Report: 01**

**Title: Scan Conversion**

*Course title: Computer Graphics Laboratory*

*Course code: CSE-304*

*3rd Year 1st Semester Examination 2022*

**Date of Submission**: 28/05/2023

****

**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 |
| 366 |  | Lamia Binta Latif |

**Experiment no-01: Scan conversion of a point**

**Source code:**

#include <iostream>

using namespace std;

char screen[80][25] = {};

void scanConvertPoint(int x, int y)

{

screen[x][y] = '\*';

}

int main()

{

int x = 40;

int y = 12;

scanConvertPoint(x, y);

for (int j = 0; j < 25; j++) {

for (int i = 0; i < 80; i++) {

cout << screen[i][j];

}

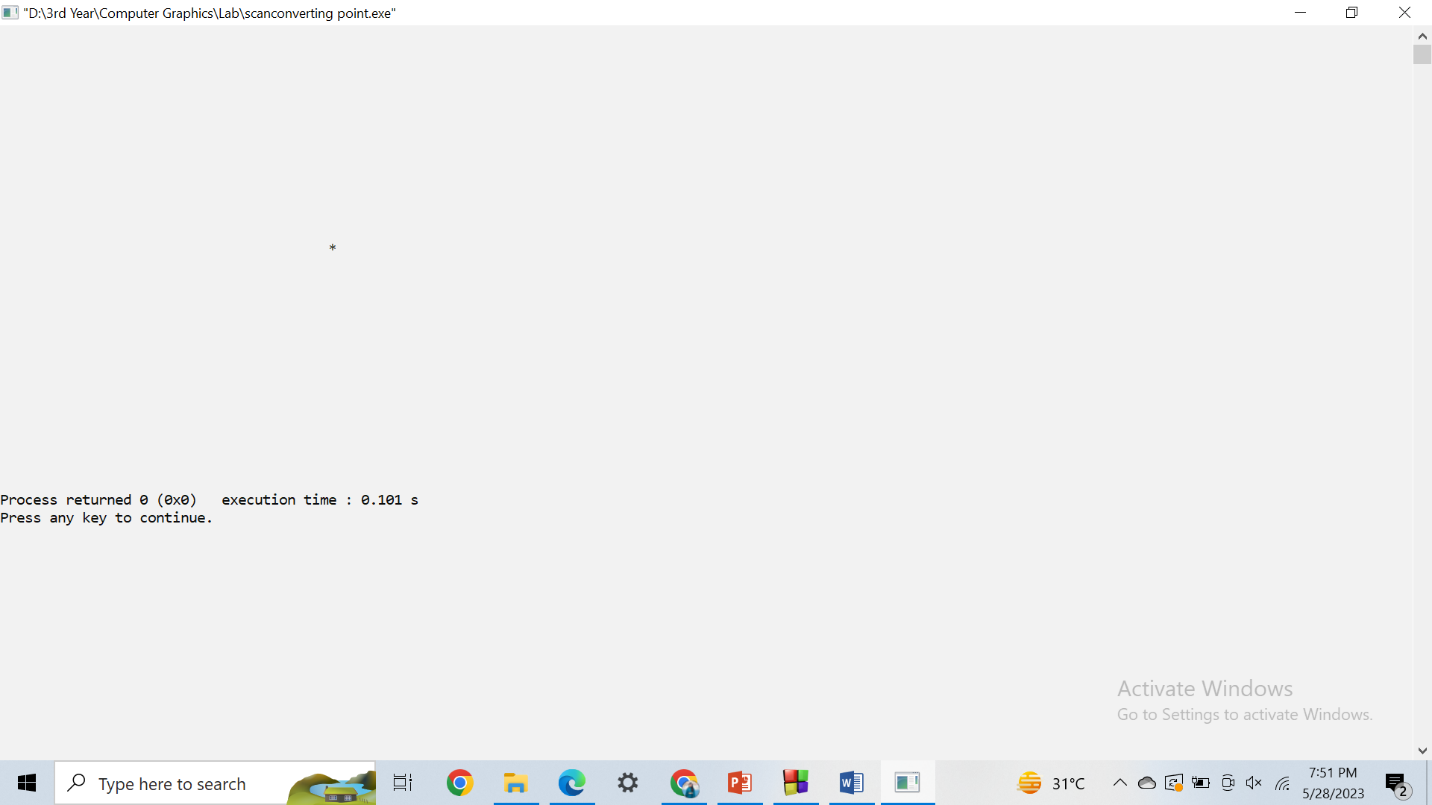
cout << endl;

}

return 0;

}

**Output:**



**Experiment no-02: Scan conversion of Line using DDA Algorithm**

**Source code:**

#include <iostream>

#include <bits/stdc++.h>

using namespace std;

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

{

int dx = x2 - x1;

int dy = y2 - y1;

int steps = abs(dx) > abs(dy) ? abs(dx) : abs(dy);

float xIncrement = static\_cast<float>(dx) / steps;

float yIncrement = static\_cast<float>(dy) / steps;

float x = x1;

float y = y1;

for (int i = 0; i <= steps; i++) {

int roundedX = static\_cast<int>(round(x));

int roundedY = static\_cast<int>(round(y));

cout << "(" << roundedX << ", " << roundedY << ")" << endl;

x += xIncrement;

y += yIncrement;

}

}

int main()

{

int x1 = 1;

int y1 = 1;

int x2 = 8;

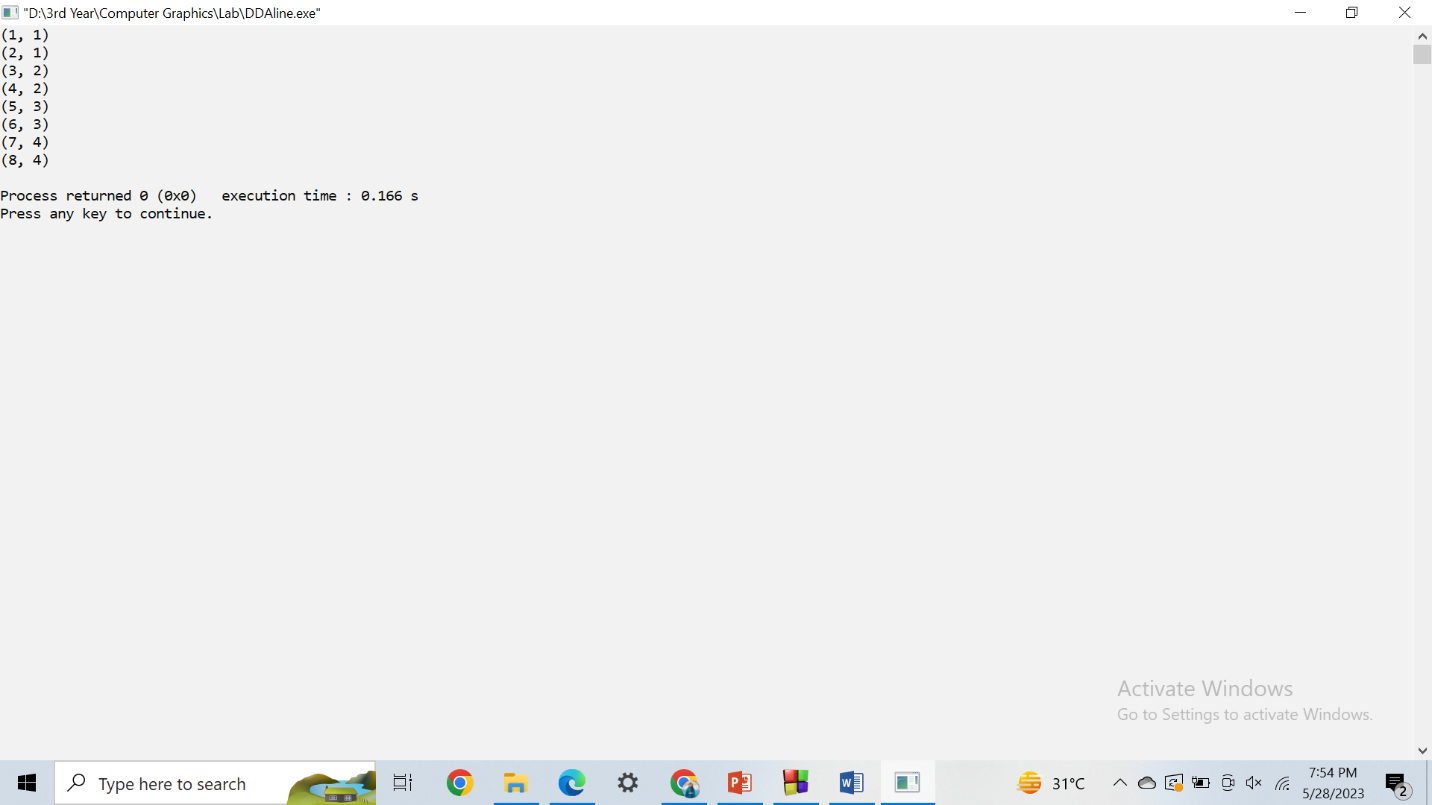
int y2 = 4;

scanConvertLine(x1, y1, x2, y2);

return 0;

}

**Output:**



**Experiment no-03: Scan conversion of Line using Bresenham Algorithm**

**Source code:**

#include <iostream>

#include <bits/stdc++.h>

using namespace std;

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

{

int dx = abs(x2 - x1);

int dy = abs(y2 - y1);

int dx2 = dx \* 2;

int dy2 = dy \* 2;

int ix = (x1 < x2) ? 1 : -1;

int iy = (y1 < y2) ? 1 : -1;

int x = x1;

int y = y1;

if (dx > dy) {

int error = dy2 - dx;

while (x != x2) {

std::cout << "(" << x << ", " << y << ")" << std::endl;

if (error >= 0) {

y += iy;

error -= dx2;

}

error += dy2;

x += ix;

}

} else {

int error = dx2 - dy;

while (y != y2) {

std::cout << "(" << x << ", " << y << ")" << std::endl;

if (error >= 0) {

x += ix;

error -= dy2;

}

error += dx2;

y += iy;

}

}

cout << "(" << x2 << ", " << y2 << ")" << endl;

}

int main()

{

int x1 = 1;

int y1 = 1;

int x2 = 8;

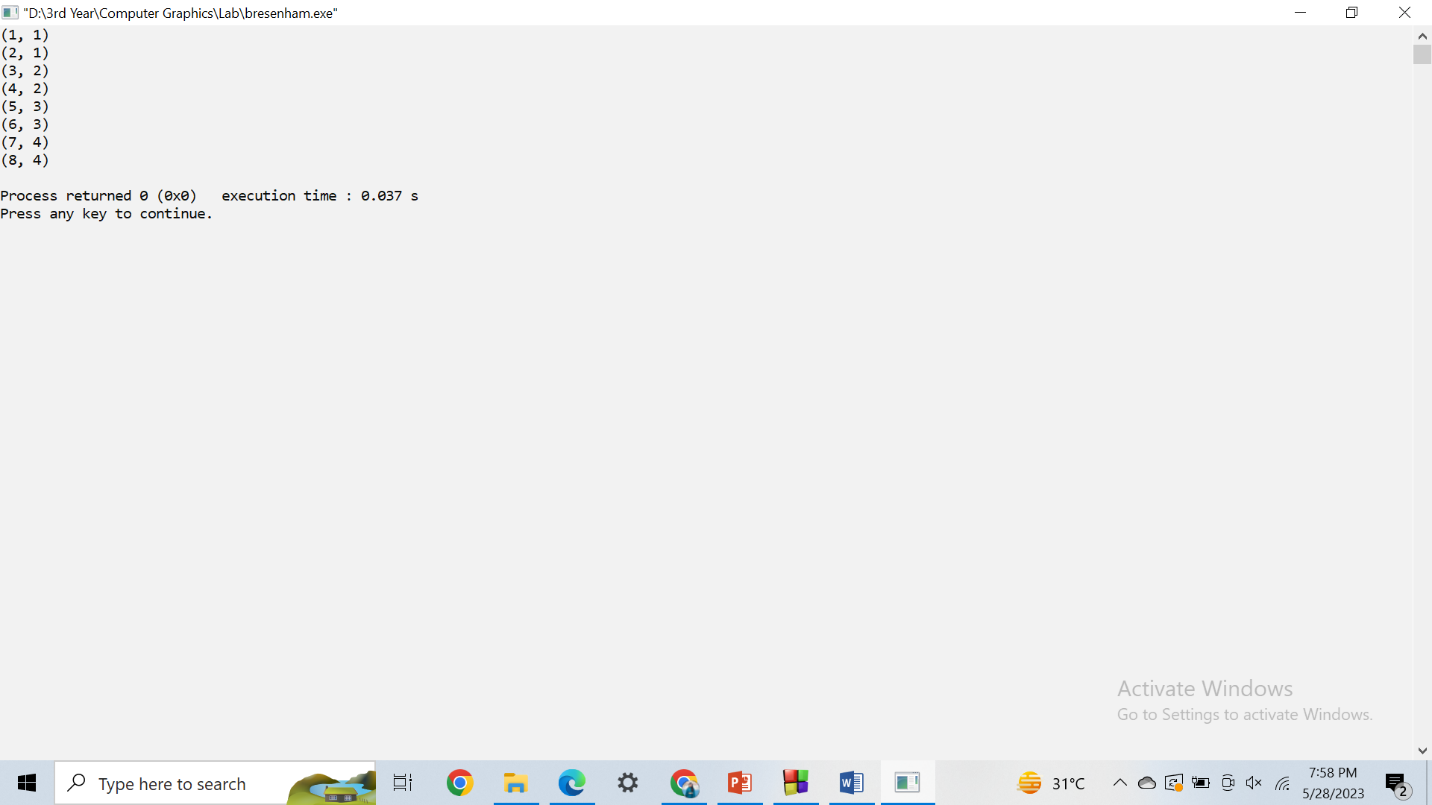
int y2 = 4;

scanConvertLine(x1, y1, x2, y2);

return 0;

}

**Output:**



**Experiment no-04: Scan conversion of Circle using Bresenham Circle Algorithm**

**Source code:**

#include <iostream>

using namespace std;

void drawCircle(int radius)

{

int x = 0;

int y = radius;

int d = 3 - 2 \* radius;

while (x <= y)

{

std::cout << "(" << x << ", " << y << ")" << std::endl;

if (d < 0)

{

d = d + 4 \* x + 6;

}

else

{

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

y--;

}

x++;

}

}

int main()

{

int radius;

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

cin >> radius;

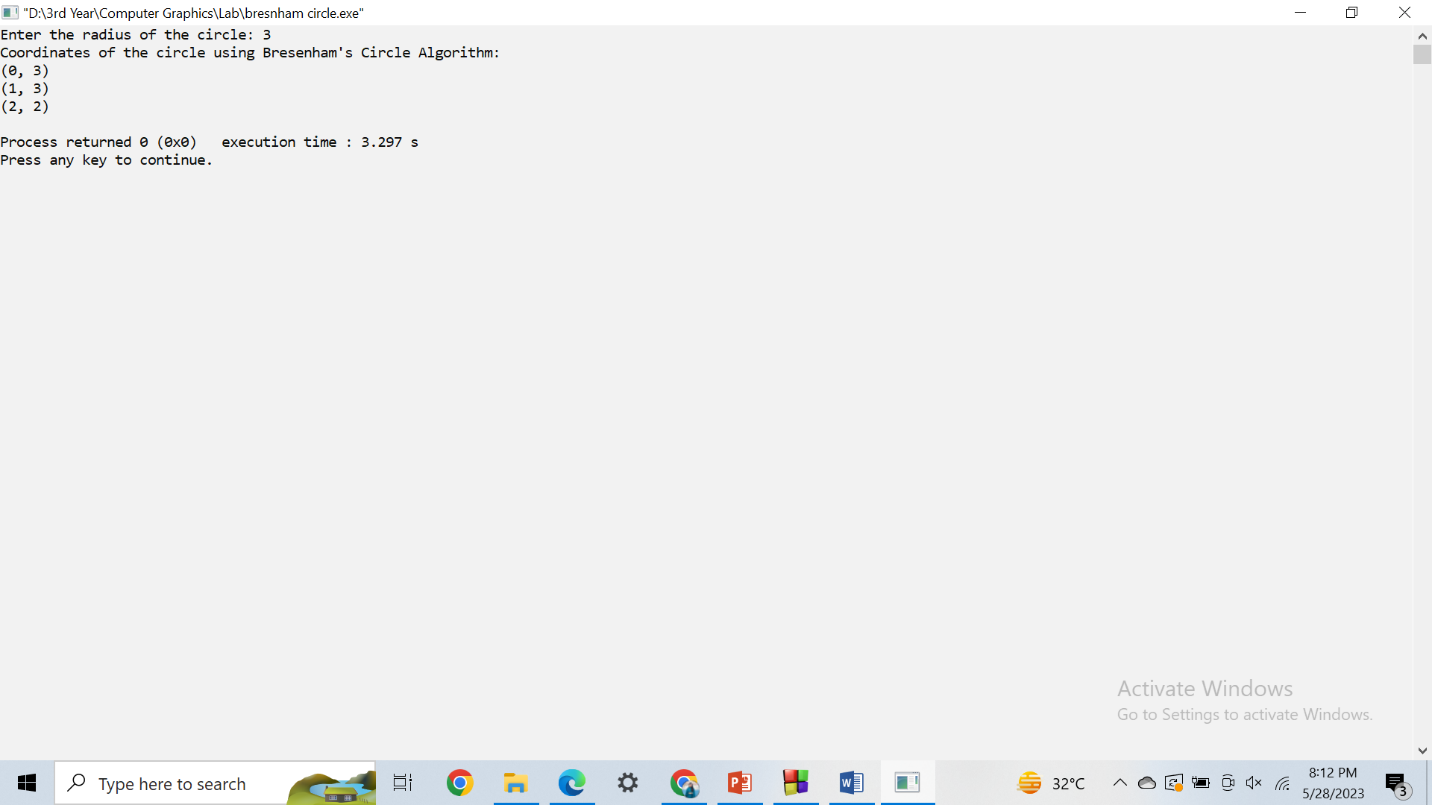
cout << "Coordinates of the circle using Bresenham's Circle Algorithm:" <<endl;

drawCircle(radius);

return 0;

}

**Output:**

****