**Title: Lab Report No.1**

*Course title: Computer Graphics Laboratory*

*Course code: CSE-304*

*3rd Year 1st Semester Examination 2022*

**Date of Submission**: 28 May 2023

****

**Submitted to-**

**Dr. Mohammad Shorif Uddin**

*Professor*

*Department of Computer Science and Engineering*

*Jahangirnagar University*

*Savar, Dhaka-1342*

**Dr. Morium Akter**

*Associate Professor*

*Department of Computer Science and Engineering*

*Jahangirnagar University*

*Savar, Dhaka-1342*

| **Sl** | Class Roll | Exam Roll | Name |
| --- | --- | --- | --- |
| 01 | 389 | 202201 | Md.Toufik Hasan Labib |

**Scan Convert a Point:**

#include <graphics.h>

#include <iostream>

#include <conio.h>

using namespace std;

int main()

{

int gd = DETECT, gm;

int a, b;

cout << "Enter the x-coordinate: ";

cin >> a;

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

cin >> b;

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

putpixel(a, b, WHITE);

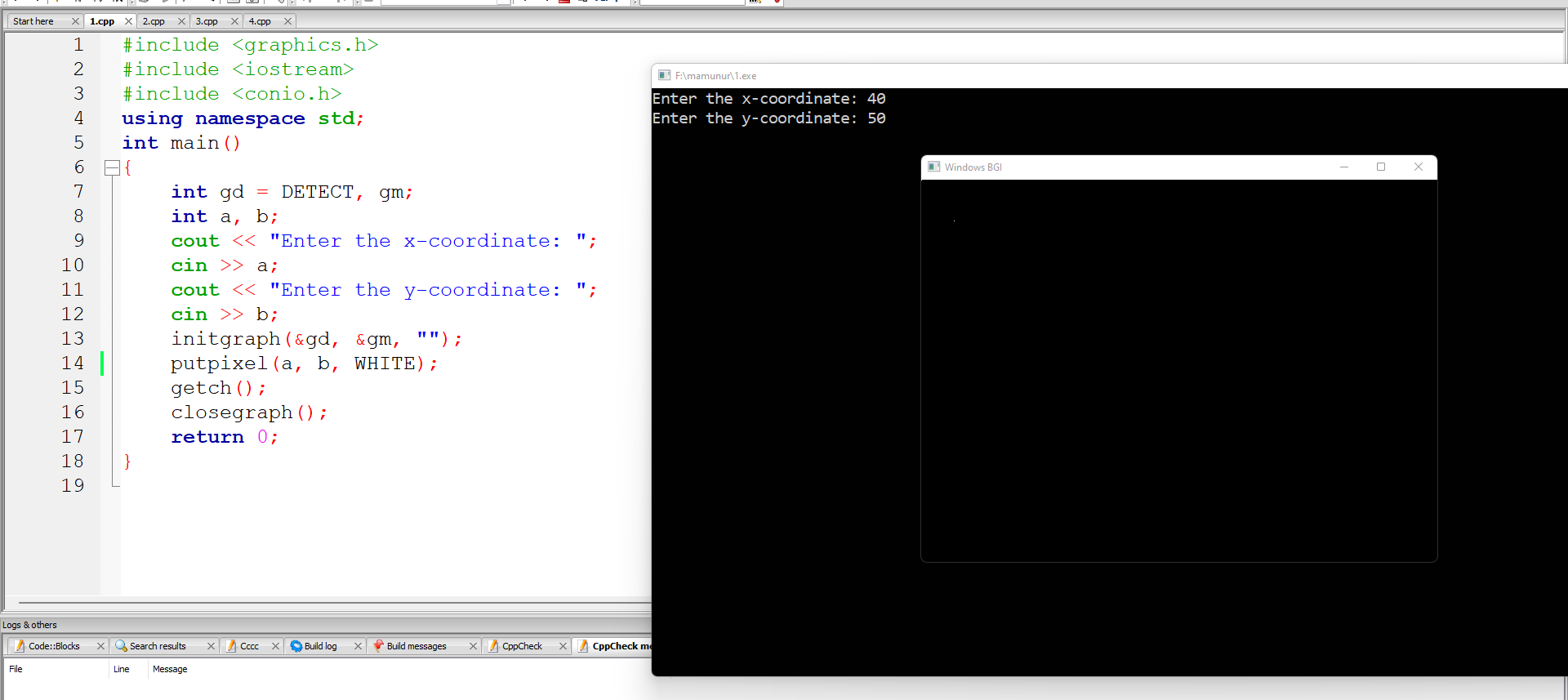
getch();

closegraph();

return 0;

}

**Output:**



**Scan Convert a Line (DDA Algorithm):**

**Source Code:**

#include <graphics.h>

#include <iostream>

#include <conio.h>

using namespace std;

int main() {

int gd = DETECT, gm;

int x1, y1, x2, y2;

int steps, xinc, yinc, dx, dy;

cout << "Enter the starting coordinates:";

cin >> x1 >> y1;

cout << "Enter the ending coordinates:";

cin >> x2 >> y2;

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

dx = x2 - x1;

dy = y2 - y1;

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

steps = abs(dx);

} else {

steps = abs(dy);

}

xinc = dx / steps;

yinc = dy / steps;

for (int i = 1; i <= steps; i++) {

putpixel(x1, y1, CYAN);

delay(10);

x1 = x1 + xinc;

y1 = y1 + yinc;

}

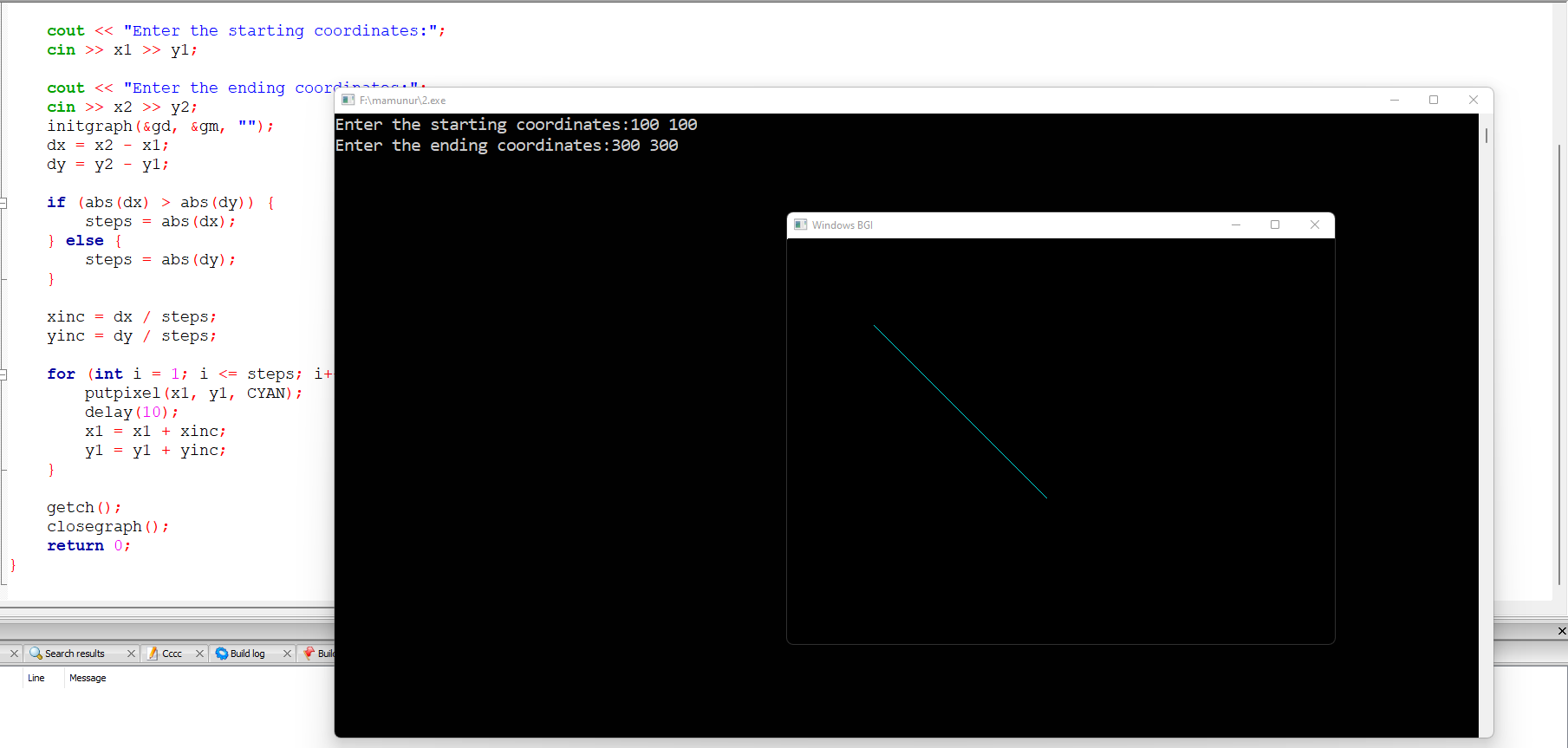
getch();

closegraph();

return 0;

}

**Output:**

****

**Scan Convert a Line (Bresenham’s Algorithm):**

**Source Code:**

#include <graphics.h>

#include <iostream>

#include <conio.h>

using namespace std;

void drawline(int x0, int y0, int x1, int y1) {

int dx, dy, p, x, y;

dx = x1 - x0;

dy = y1 - y0;

x = x0;

y = y0;

p = 2 \* dy - dx;

while (x < x1) {

if (p >= 0) {

putpixel(x, y, BLUE);

y = y + 1;

p = p + 2 \* dy - 2 \* dx;

} else {

putpixel(x, y, BLUE);

p = p + 2 \* dy;

}

x = x + 1;

delay(10);

}

}

int main() {

int gd = DETECT, gm, error, x0, y0, x1, y1;

cout << "Enter coordinates of first point:";

cin >> x0 >> y0;

cout << "Enter coordinates of second point:";

cin >> x1 >> y1;

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

drawline(x0, y0, x1, y1);

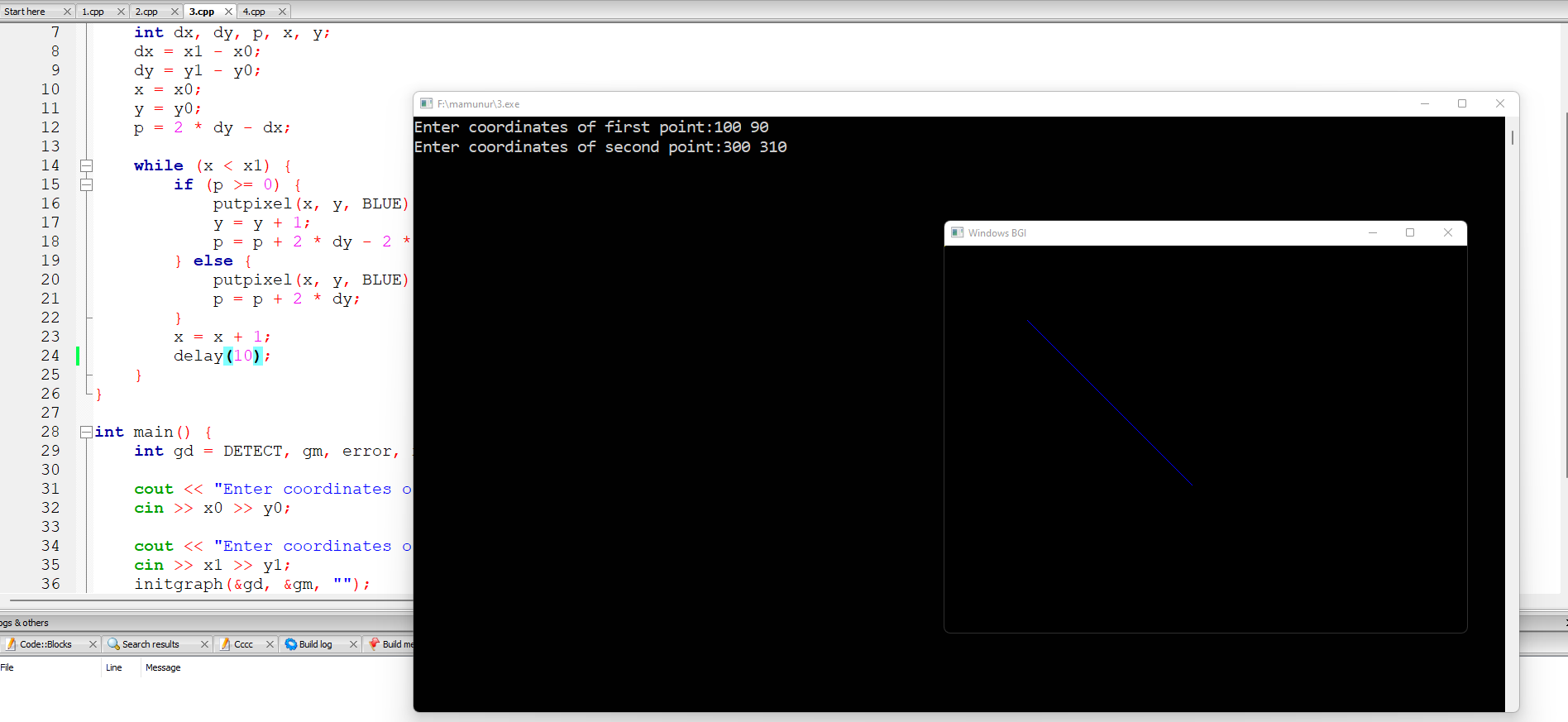
getch();

closegraph();

return 0;

}

**Output:**



**Scan Convert a Circle (Bresenham’s Circle Algorithm):**

**Source Code:**

#include <stdio.h>

#include <dos.h>

#include <graphics.h>

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

{

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

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

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

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

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

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

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

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

}

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);

}

}

int main()

{

int xc = 200, yc = 200, r = 100;

int gd = DETECT, gm;

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

circleBres(xc, yc, r);

getch();

closegraph();

return 0;

}

**Output:**

