

Lab Report. 01

Title: Lab Report

Course title: Computer Graphics Lab

Course code: CSE-304

3rd Year 1st Semester 2022

Date of Submission: 28/05/2023



Submitted to-

Dr. Mohammad Shorif Uddin

Professor

Department of Computer Science and Engineering

Jahangirnagar University

Savar, Dhaka-1342

And

Dr. Morium Akter

Associate Professor

Department of Computer Science and Engineering

Jahangirnagar University

Savar, Dhaka-1342

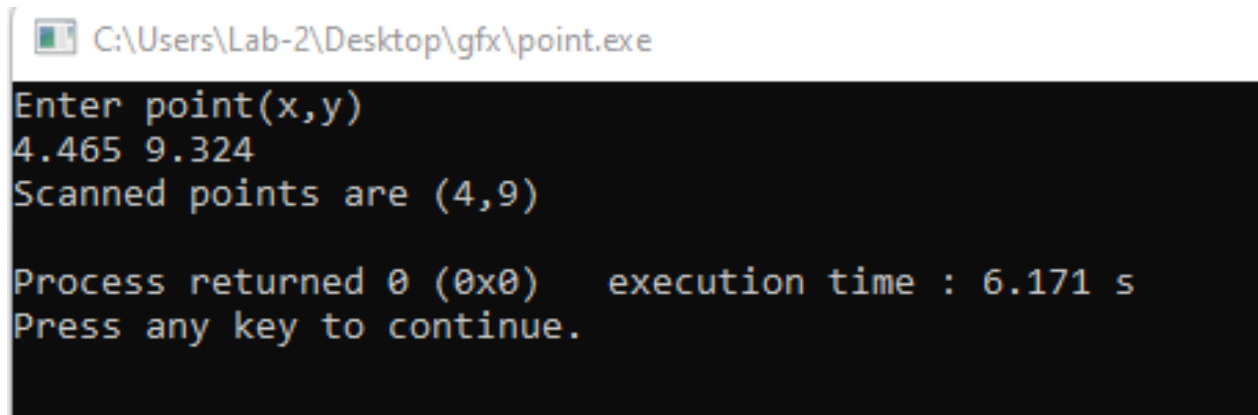
SI	Class Roll	Exam Roll	Name
01	390		Md Abdullah Al Mamun

Scan conversion of a point:

Source code:

```
#include<bits/stdc++.h>
using namespace std;
int main()
{
float x,y;
cout << "Enter point(x,y) " << endl;
cin >> x>>y;
int ax,by;
ax = floor(x);
by = floor(y);
cout << "Scanned points are
("<<ax<<","<<by<<") "<<endl;
}
```

Output:



```
C:\Users\Lab-2\Desktop\gfx\point.exe
Enter point(x,y)
4.465 9.324
Scanned points are (4,9)

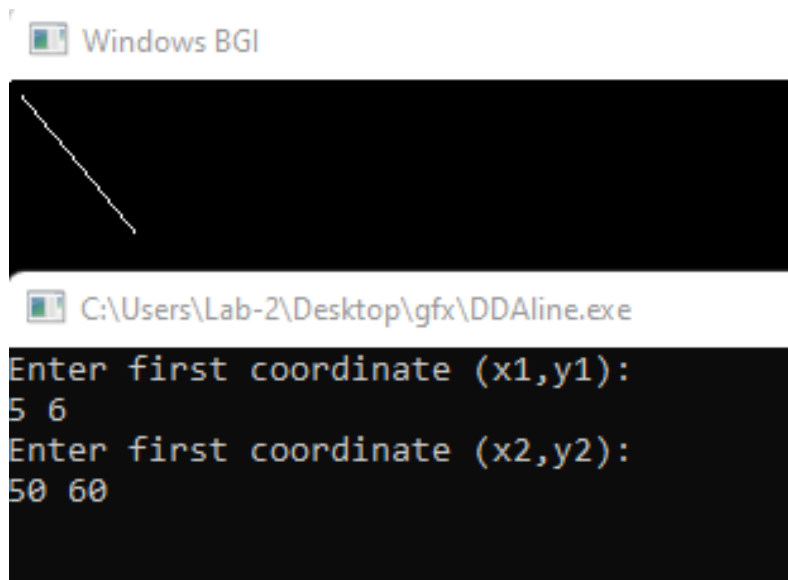
Process returned 0 (0x0)   execution time : 6.171 s
Press any key to continue.
```

Scan converting a line using DDA algorithm:

Source code:

<pre>#include<bits/stdc++.h> #include<graphics.h> using namespace std; void drawlinedda(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++) { putpixel(static_cast<int>(x),static _cast<int>(y),WHITE); x += xincrement; y += yincrement; } }</pre>	<pre>int main() { int gd = DETECT, gm; initgraph(&gd,&gm, ""); int x1,y1,x2,y2; cout << "Enter first coordinate (x1,y1):" << endl; cin >> x1 >> y1; cout << "Enter first coordinate (x2,y2):" << endl; cin >> x2>>y2; drawlinedda(x1,y1,x2,y2); delay(50000000); closegraph(); return 0; }</pre>
---	--

Output:

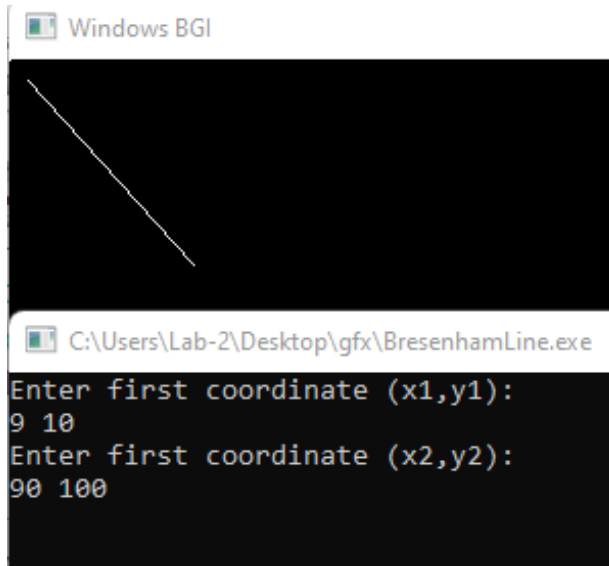


Scan Converting a line using Bransenham Algorithm

Source code:

<pre>#include<bits/stdc++.h> #include<graphics.h> using namespace std; void drawlinebresenham(int x1, int y1, int x2, int y2) { int dx = abs(x2 - x1); int dy = abs(y2- y1); int sx = (x1 < x2)? 1 : -1; int sy = (y1 < y2)? 1 : -1; int err = dx - dy; while(true) { putpixel(x1,y1,WHITE); if(x1==x2 && y1==y2) break; int dt = 2*err; if(dt>-dy) { err -= dy; x1 += sx; } } }</pre>	<pre>if(dt < dx) { err += dx; y1 += sy; } } int main() { int gd = DETECT, gm; initgraph(&gd,&gm, ""); int x1,y1,x2,y2; cout << "Enter first coordinate (x1,y1):" << endl; cin >> x1 >> y1; cout << "Enter first coordinate (x2,y2):" << endl; cin >> x2>>y2; drawlinebresenham(x1,y1,x2,y2); delay(50000000); closegraph(); return 0; }</pre>
---	---

Output:



Scan converting a circle using Brasenham circle algorithm:

Source code:

<pre> #include <iostream> #include <cmath> #include <graphics.h> using namespace std; void drawCircleBresenham(int xc, int yc, int radius) { int x = 0; int y = radius; int d = 3 - 2 * 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 (d < 0) { d += 4 * x + 6; } } </pre>	<pre> else { d += 4 * (x - y) + 10; 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; drawCircleBresenham(xc, yc, radius); delay(5000); closegraph(); return 0; } </pre>
--	---

Output:

