

Lab Report 1

Course title: Computer Graphics Laboratory

Course code: CSE-304

3rd Year 1st Semester

Date of Submission: 28/5/2023

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	405	202217	Farhan Ahmed Onu

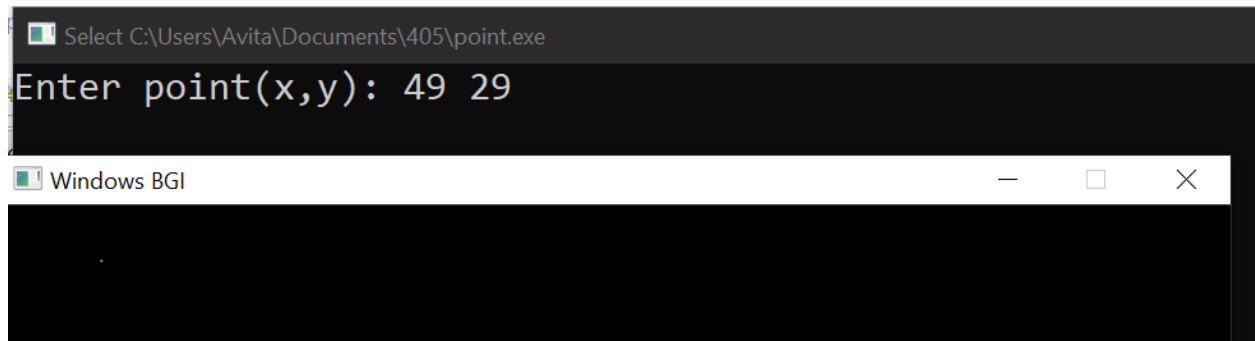
Scan a Point:

Source code:

```
#include<bits/stdc++.h>
#include<graphics.h>

int main()
{
    float x,y;
    int x1,y1;
    printf("Enter point(x,y): ");
    scanf("%f %f",&x,&y);
    x1 = floor(x);
    y1 = floor(y);
    int gd= DETECT, gm;
    initgraph(&gd,&gm,"");
    putpixel(x1,y1,WHITE);
    getch();
    closegraph();
    return 0;
}
```

Screenshot:



Scanning a line using DDA algorithm:

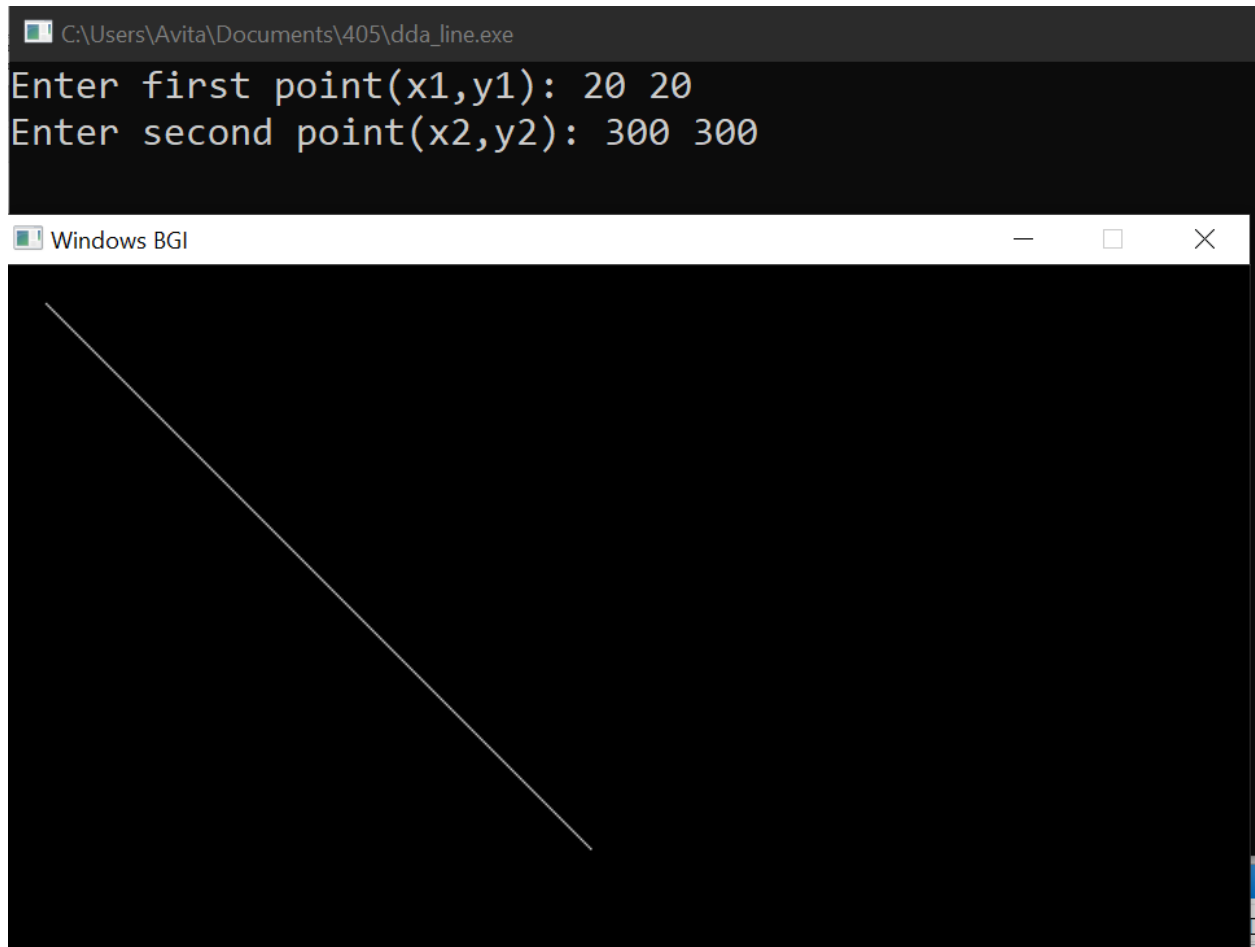
Source Code:

```
#include<bits/stdc++.h>
#include<graphics.h>

int main()
{
    float x1,y1,x2,y2,m,b;
    int sx1,sy1,sx2,sy2;
    printf("Enter first point(x1,y1): ");
    scanf("%f %f",&x1,&y1);
    printf("Enter second point(x2,y2): ");
    scanf("%f %f",&x2,&y2);
    sx1 = floor(x1);
    sy1 = floor(y1);
    sx2 = floor(x2);
    sy2 = floor(y2);
    m = (sy2-sy1)/(sx2-sx1);
    b = sy1 - (m * sx1);
    int steps,k;
    float x_inc,y_inc;
    if(abs(sx2-sx1) > abs(sy2-sy1))
    {
        steps=abs(sx2-sx1);
    }
    else
    {
        steps=abs(sy2-sy1);
    }
    x_inc = (sx2-sx1) / (float)steps;
    y_inc = (sy2-sy1) / (float)steps;
    int gd= DETECT, gm;
    initgraph(&gd,&gm,"");
    putpixel(sx1,sy1,WHITE);
    for(k = 0;k < steps;k++)
    {
        sx1 += x_inc;
        sy1 += y_inc;
        putpixel(round(sx1),round(sy1),WHITE);
    }
    getch();
    closegraph();
}
```

```
    return 0;  
}
```

Screenshot:



Scanning a line using bresenham line algorithm:

Source code:

```
#include <iostream>
#include <graphics.h>
using namespace std;

int main()
{
    int gd = DETECT, gm;
    initgraph(&gd, &gm, "");

    int x1, y1, x2, y2;

    cout << "Enter the 1st point (x1, y1): ";
    cin >> x1 >> y1;

    cout << "Enter the 2nd point (x2, y2): ";
    cin >> x2 >> y2;

    int dx = abs(x2 - x1);
    int dy = abs(y2 - y1);
    int x, y;

    if (x1 < x2)
    {
        x = x1;
        y = y1;
    }
    else
    {
        x = x2;
        y = y2;
        x2 = x1;
        y2 = y1;
    }

    int p = 2 * dy - dx;
    putpixel(x, y, WHITE);

    while (x < x2)
    {
        x++;
```

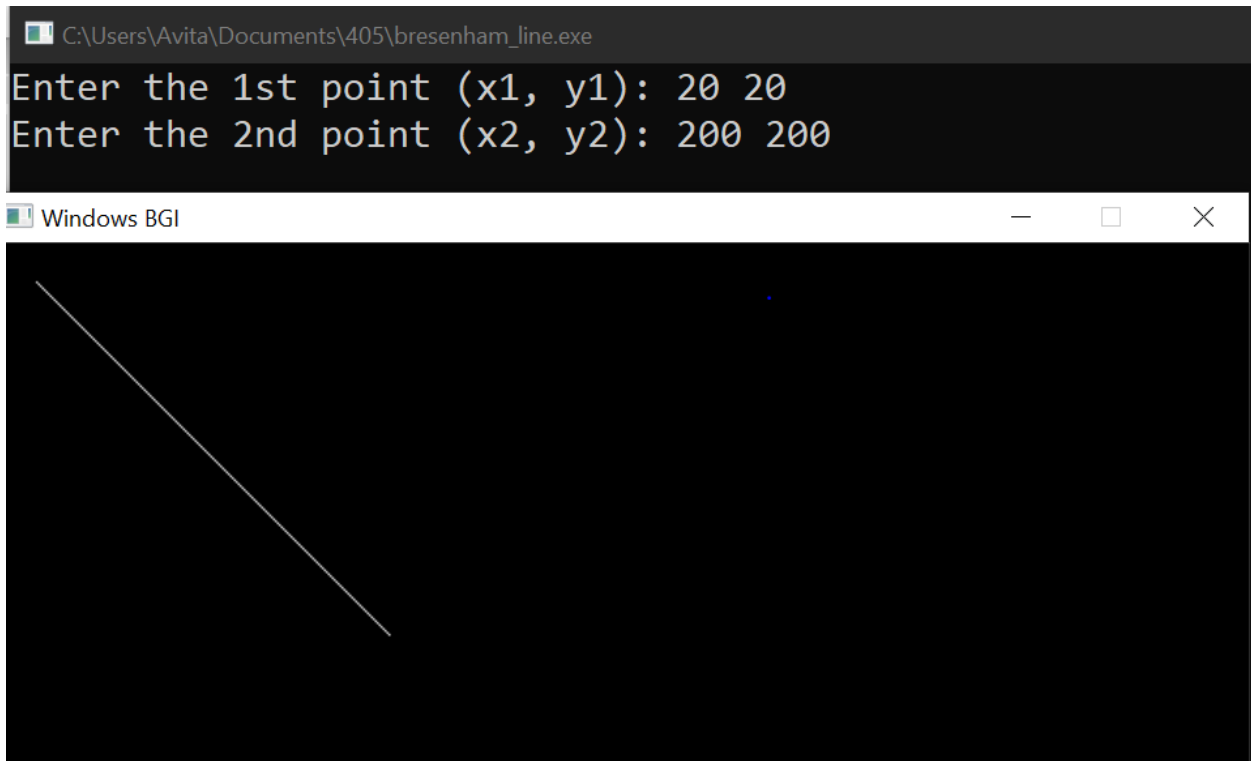
```
        if (p < 0)
        {
            p += 2 * dy;
        }
        else
        {
            y++;
            p += 2 * (dy - dx);
        }

        putpixel(x, y, WHITE);
    }

    getch();
    closegraph();

    return 0;
}
```

Screenshot:



Scanning a circle using bresenham circle algorithm:

Source code:

```
#include <iostream>
#include <graphics.h>
using namespace std;

int main()
{
    int gd = DETECT, gm;
    initgraph(&gd, &gm, "");

    int xc, yc, radius;

    cout << "Enter the center position: ";
    cin >> xc >> yc;

    cout << "Enter the radius of the circle: ";
    cin >> 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 = d + 4 * x + 6;
        }
        else
        {
            d = d + 4 * (x - y) + 10;
            y--;
        }
    }
}
```

```
    }  
  
    x++;  
}  
  
getch();  
closegraph();  
  
return 0;  
}
```

Screenshot:

