LAB Assignment-03

Course title: Computer Graphics Laboratory
Course code: CSE-304
3nd Year 1st Semester Examination 2022

Date of Submission: 11 June,2023

Submitted to-

Dr. Mohammad Shorif Uddin
Professor

Dr. Morium Akter
Associate Professor

Department of Computer Science and Engineering
Jahangirnagar University
Savar, Dhaka-1342

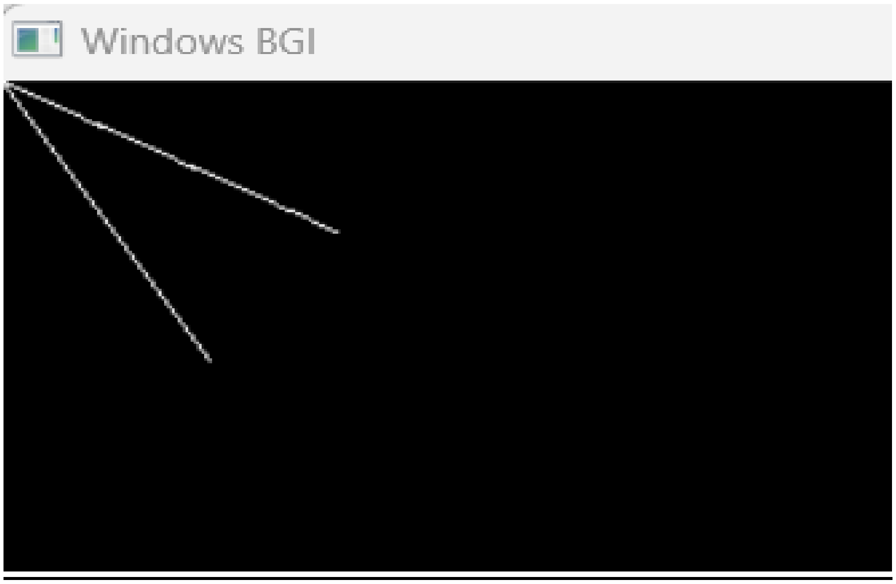
Sl	Class Roll	Exam Roll	Name
01	386	202198	Sunirmol Mollik

Labwork:01 Scan converted of a line object from(0,0) to(100,150) and rotted it by 30 degree.

Source code:

```
#include <graphics.h>
#include <iostream>
#include <cmath>

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

    int x1 = 0, y1 = 0, x2 = 100, y2 = 50;

    // Original line
    line(x1, y1, x2, y2);

    // Rotate by 30 degrees
    float theta = 30 * (M_PI / 180);

// Convert degrees to radians
    int new_x1 = round(x1 * cos(theta) - y1 * sin(theta));
    int new_y1 = round(x1 * sin(theta) + y1 * cos(theta));
    int new_x2 = round(x2 * cos(theta) - y2 * sin(theta));
    int new_y2 = round(x2 * sin(theta) + y2 * cos(theta));

    // Rotated line
    line(new_x1, new_y1, new_x2, new_y2);
delay(5000*3600);
    closegraph();
    return 0;
}
```

Output:



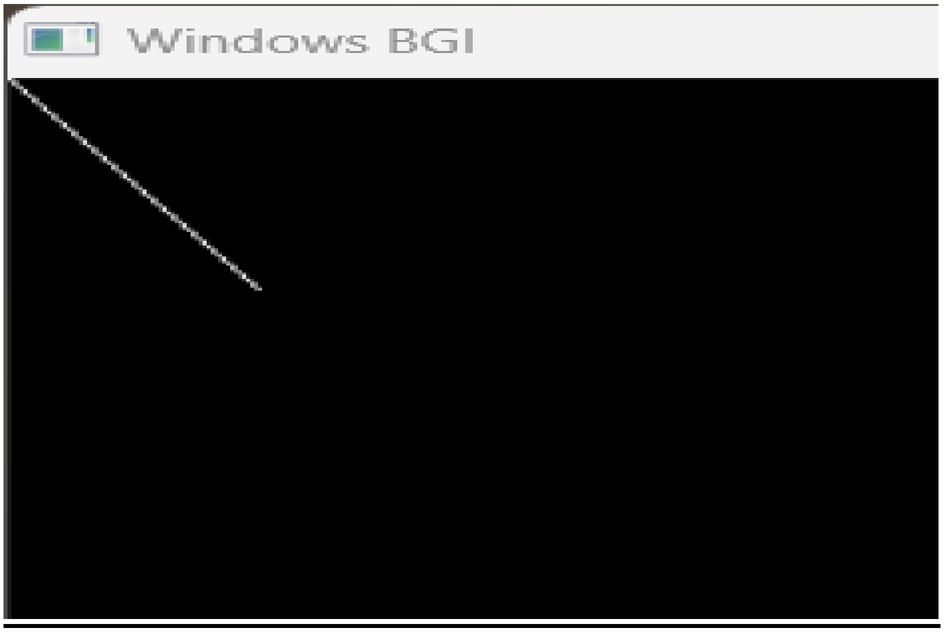
Labwork:02 Scan converted of a line object from(0,0) to(100,150) and scale it to 50%.

Source Code:

```
#include <graphics.h>

void drawLine(int x0, int y0, int x1, int y1)
{
    int dx = abs(x1 - x0);
    int dy = abs(y1 - y0);
    int sx = (x0 < x1) ? 1 : -1;
    int sy = (y0 < y1) ? 1 : -1;
    int err = dx - dy;

    while (true)
    {
        putpixel(x0, y0, WHITE);

        if (x0 == x1 && y0 == y1)
        {
            break;
        }

        int e2 = 2 * err;
        if (e2 > -dy)
        {
            err -= dy;
            x0 += sx;
        }
        if (e2 < dx)
{
            err += dx;
            y0 += sy;
        }
    }
}
```

```
void scaleLine(int& x0, int& y0, int& x1, int& y1, double scaleFactor)
{
    x0 = static_cast<int>(x0 * scaleFactor);
    y0 = static_cast<int>(y0 * scaleFactor);
    x1 = static_cast<int>(x1 * scaleFactor);
    y1 = static_cast<int>(y1 * scaleFactor);
}

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

    int x0 = 0;
    int y0 = 0;
    int x1 = 100;
    int y1 = 150;

    // Scale to 50%
    double scaleFactor = 0.5;
    scaleLine(x0, y0, x1, y1, scaleFactor);

    drawLine(x0, y0, x1, y1);
getch();
    closegraph();

    return 0;
}
```

Output:

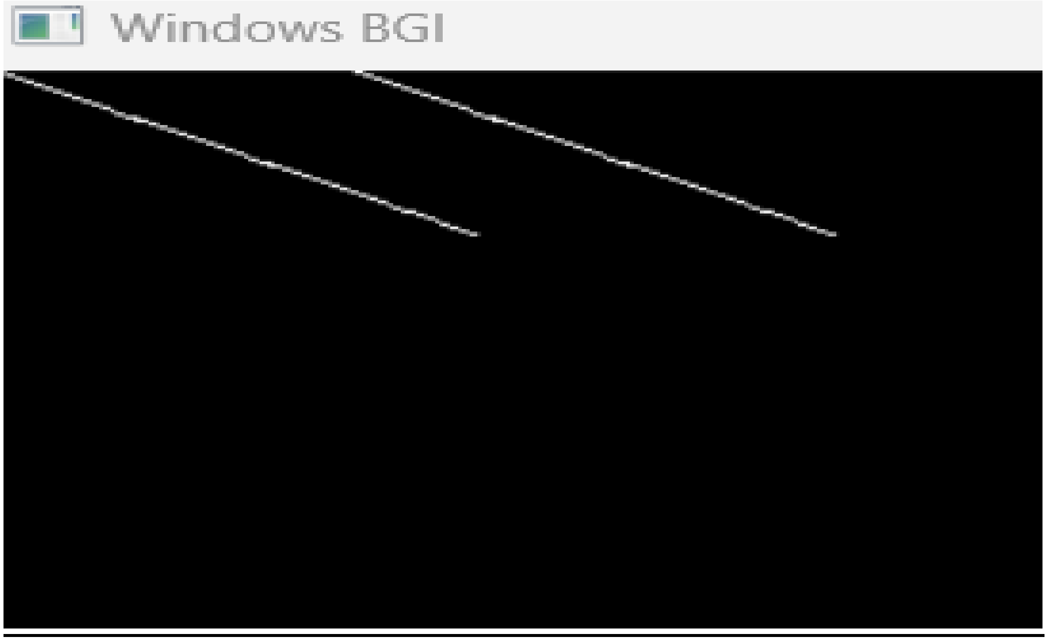


Labwork:03 Scan converted of a line object from(0,0) to(100,150) and and translate it on x axis by 75 pixels.

Source Code:

```
#include <graphics.h>
#include <iostream>

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

    int x1 = 0, y1 = 0, x2 = 100, y2 = 50;

    // Original line
    line(x1, y1, x2, y2);

int translate_x = 75;
    int new_x1 = x1 + translate_x;
    int new_y1 = y1;
    int new_x2 = x2 + translate_x;
    int new_y2 = y2;

    // Translated line
    line(new_x1, new_y1, new_x2,
new_y2);

    delay(5000);
    closegraph();
    return 0;
}
```

Output:



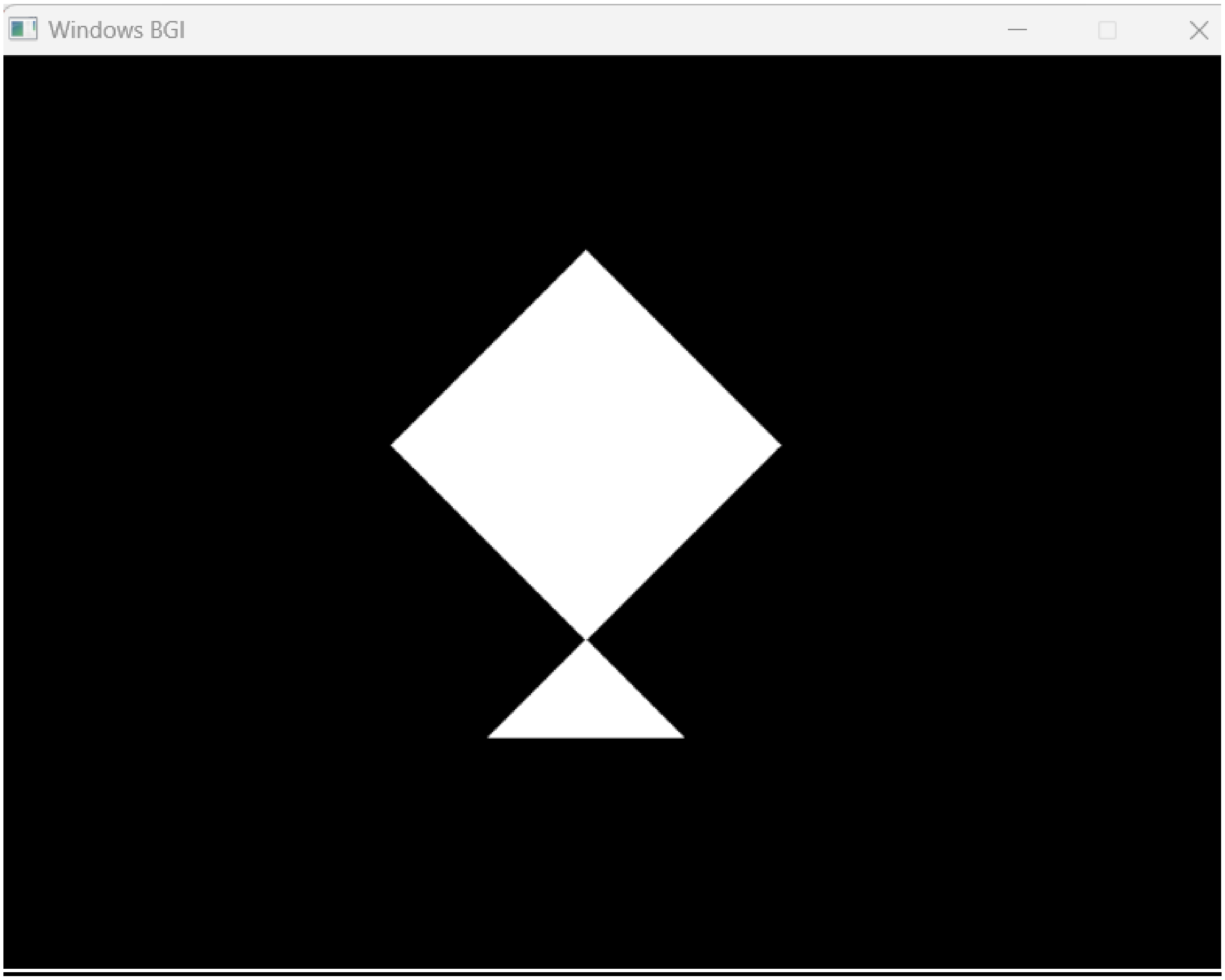
Labwork:04 Drawing a kite using Brasenham line Algorithm.

Source Code:

```
include <graphics.h>
#include <iostream>
#include <conio.h>
#include <math.h>

using namespace std;
void line_bressenham(int x1,int y1,int
x2,int 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, BLACK);
    }
}
```

```
void kite()
{
    line(200, 200, 300, 100);
    line(300, 100, 400, 200);
    line(400, 200, 300, 300);
    line(300, 100, 300, 300);
    line(300,300,200,200);

    //arc(300, 300, 45, 135, 140);
    setfillstyle(SOLID_FILL, 12);

    floodfill(301, 105, WHITE);
    setfillstyle(SOLID_FILL, 12);
floodfill(299, 105, WHITE);
    setfillstyle(SOLID_FILL, WHITE);

    floodfill(299, 275, WHITE);
    setfillstyle(SOLID_FILL, WHITE);

    floodfill(301, 275, WHITE);
    line(300, 300, 250, 350);
    line(250, 350, 350, 350);
    line(300, 300, 350, 350);
//    line_bressenham(300,300,250,350);
//    line_bressenham(250,350,350,350);
//    line_bressenham(300,300,350,350);
    setfillstyle(SOLID_FILL, WHITE);

    floodfill(300, 310, WHITE);
{
    int gd = DETECT, gm;
    initgraph(&gd, &gm, "");

    kite();

    getch();
    closegraph();

    return 0;
}
```

Output:


Windows BGI