**Lab Report**

*Course title: Computer Graphics*

*Course code: CSE-304*

*3rd Year 1st Semester Examination 2022*

**Date of Submission**: 11 June 2023

****

*Department of Computer Science and Engineering*

*Jahangirnagar University*

*Savar, Dhaka-1342*

|  |  |  |  |
| --- | --- | --- | --- |
| **Sl** | Class Roll | Exam Roll | Name |
| 01 | 399 | 202211 | Md Hasan Al Mamun |

1. **Scan convert a line object from (0,0) to (100,50)**

**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;

}

}

}

int main() {

int gd = DETECT, gm;

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

drawLine(0, 0, 100, 50);

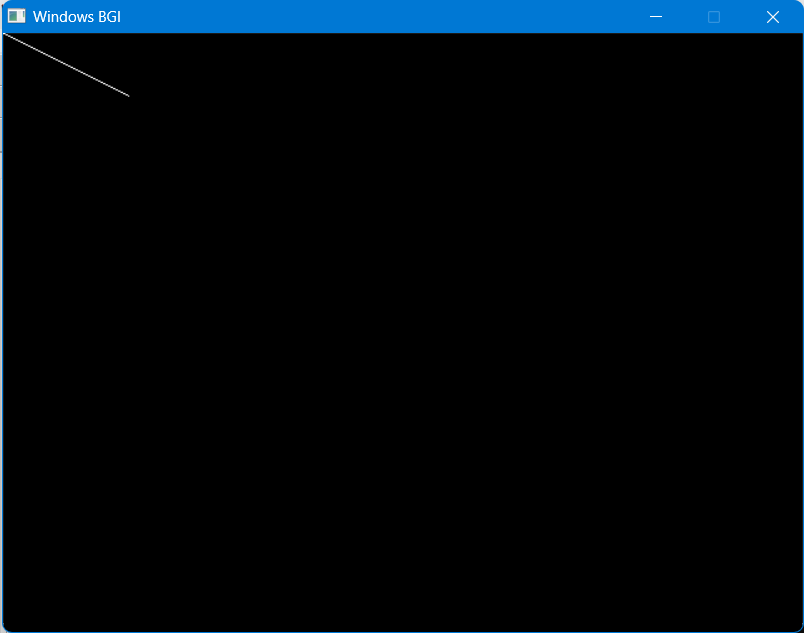
getch();

closegraph();

return 0;

}

**Output:**

****

**Rotating the Line by 30 Degree**

**Code:**

#include <graphics.h>

#include <cmath>

void rotate(int& x, int& y, double angle) {

double radians = angle \* (M\_PI / 180.0);

double cosAngle = cos(radians);

double sinAngle = sin(radians);

int newX = round(x \* cosAngle - y \* sinAngle);

int newY = round(x \* sinAngle + y \* cosAngle);

x = newX;

y = newY;

}

void drawLine(int x0, int y0, int x1, int y1, double angle) {

rotate(x0, y0, angle);

rotate(x1, y1, angle);

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;

}

}

}

int main() {

int gd = DETECT, gm;

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

drawLine(0, 0, 100, 50, 30);

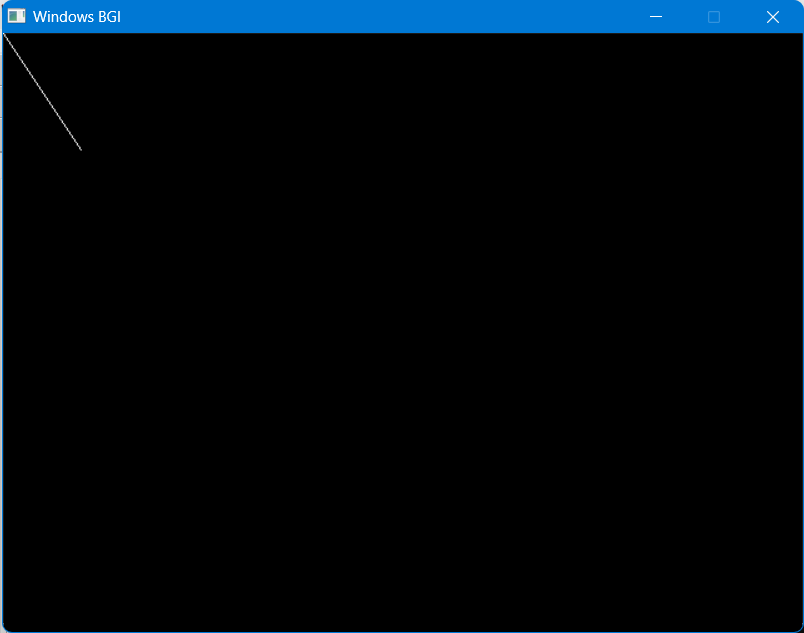
getch();

closegraph();

return 0;

}

**Output :**

****

**Scaling the Line 50%**#include <graphics.h>

#include <cmath>

void scale(int& x, int& y, double factor) {

x = round(x \* factor);

y = round(y \* factor);

}

void drawScaledLine(int x0, int y0, int x1, int y1, double scale\_factor) {

scale(x0, y0, scale\_factor);

scale(x1, y1, scale\_factor);

line(x0, y0, x1, y1);

}

int main() {

int gd = DETECT, gm;

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

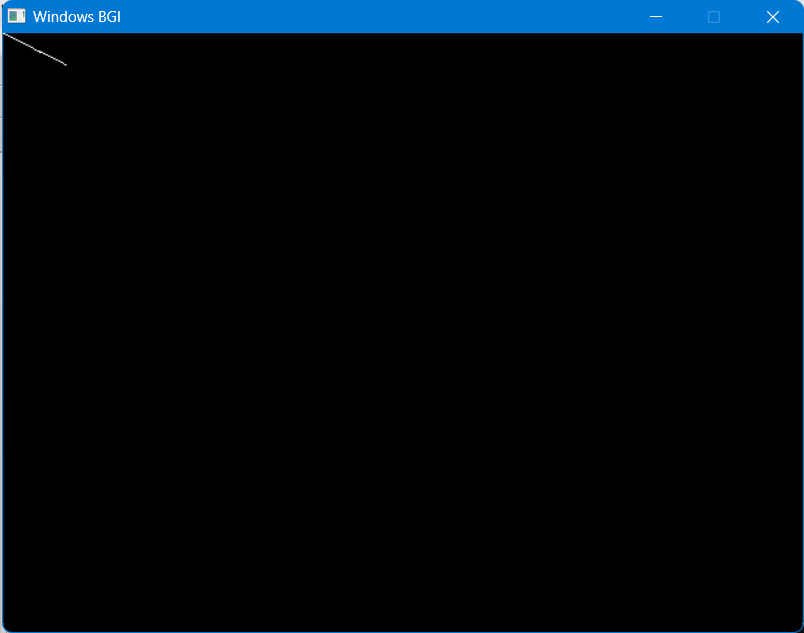
drawScaledLine(0, 0, 100, 50, 0.5);

getch();

closegraph();

return 0;

}  
  
**Output:**



**Translating the line on X Axis by 75 Pixel**

**Code:**

#include <graphics.h>

#include <cmath>

void translate(int& x, int& y, int tx, int ty) {

x += tx;

y += ty;

}

void drawTranslatedLine(int x0, int y0, int x1, int y1, int tx, int ty) {

translate(x0, y0, tx, ty);

translate(x1, y1, tx, ty);

line(x0, y0, x1, y1);

}

int main() {

int gd = DETECT, gm;

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

drawTranslatedLine(0, 0, 100, 50, 75, 0);

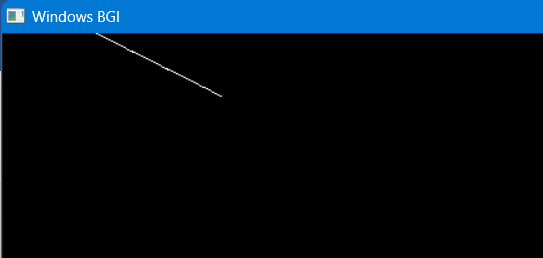
getch();

closegraph();

return 0;

}

**Output:**



**Drawing a kite Using Bresenham Line Algorithm**

**Code:**

#include <graphics.h>

void drawKite(int x, int y, int width, int height) {

int x1 = x;

int x2 = x + width / 2;

int x3 = x;

int x4 = x - width / 2;

int y1 = y - height / 2;

int y2 = y;

int y3 = y + height / 2;

int y4 = y;

// Draw the lines

line(x1, y1, x2, y2);

line(x2, y2, x3, y3);

line(x3, y3, x4, y4);

line(x4, y4, x1, y1);

}

int main() {

int gd = DETECT, gm;

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

int x = 320; // X-coordinate of the kite's center

int y = 240; // Y-coordinate of the kite's center

int width = 150; // Width of the kite

int height = 300; // Height of the kite

drawKite(x, y, width, height);

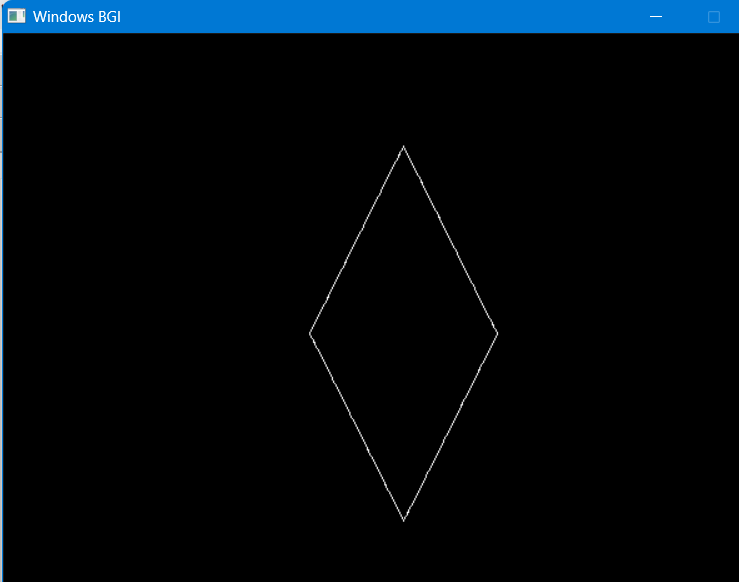
getch();

closegraph();

return 0;

}

**Output :**

****