Lab Report. 03

Title: Lab Report

Course title: Computer Graphics Lab

Course code: CSE-304

3rd Year 1st Semester 2022

Date of Submission: 11/06/2023

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

Sl	Class Roll	Exam Roll	Name
01	407	202219	Kamrul Hasan Nahid

Department of Computer Science and Engineering
Jahangirnagar University
Savar, Dhaka, Bangladesh

Experiment no: 07

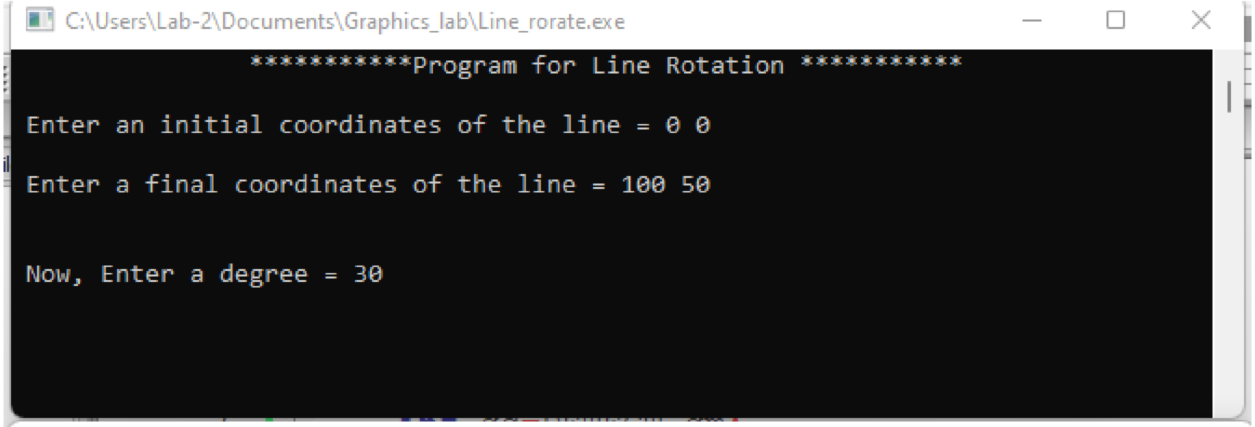
Experiment Name: Scan convert a line object from (0, 0) to (100, 50) and rotate it by 30 degree.

Source Code:

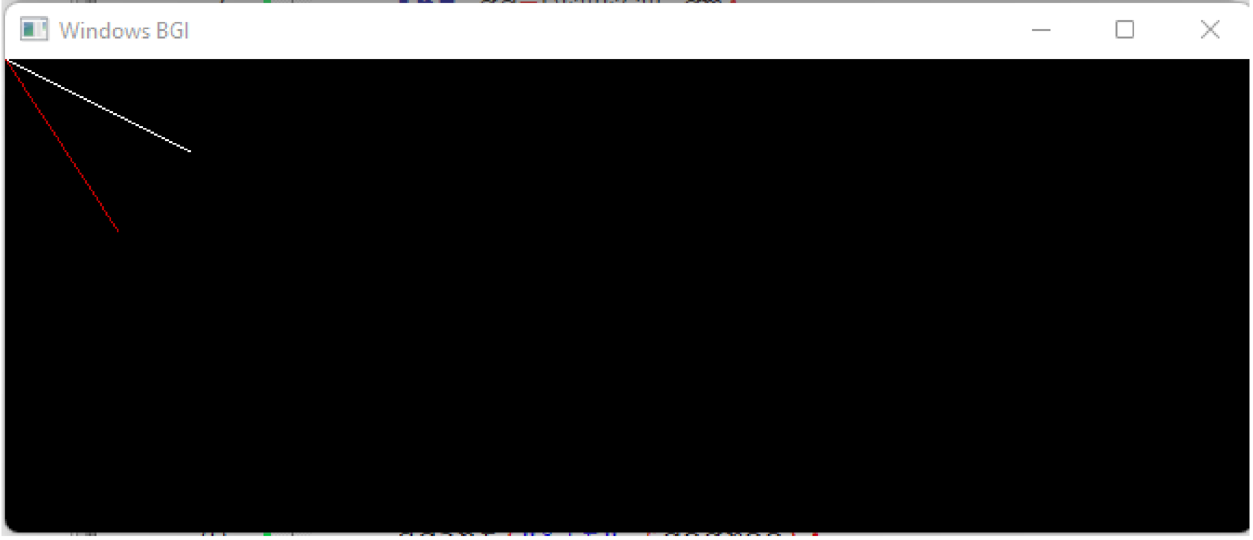
```
#include<graphics.h>
#include<stdio.h>
#include<conio.h>
#include<math.h>
int main()
{
   int gd=DETECT,gm;
   int pivot_x,pivot_y,x,y;
   double degree,radian;
   int rotated_point_x,rotated_point_y;
   initgraph(&gd,&gm,"C://TURBOC3//BGI");
   cleardevice();
   printf("\t\t***********Program for Line Rotation *********** \n");
   printf("\n Enter an initial coordinates of the line = ");
   scanf("%d %d",&pivot_x,&pivot_y);
   printf("\n Enter a final coordinates of the line = ");
   scanf("%d %d",&x,&y);
   line(pivot_x,pivot_y,x,y);
   printf("\n\n Now, Enter a degree = ");
   scanf("%lf",&degree);
   radian=degree*0.01745;
```

```
    rotated_point_x=(int)(pivot_x +((x-pivot_x)*cos(radian)-(y-pivot_y)*sin(radian)));

    rotated_point_y=(int)(pivot_y +((x-pivot_x)*sin(radian)+(y-pivot_y)*cos(radian)));

    setcolor(RED);

    line(pivot_x,pivot_y,rotated_point_x,rotated_point_y);

    getch();

    closegraph();

}
```

Input and Output:



```
                ************Program for Line Rotation ************

Enter an initial coordinates of the line = 0 0

Enter a final coordinates of the line = 100 50


Now, Enter a degree = 30
```



Experiment no: 08

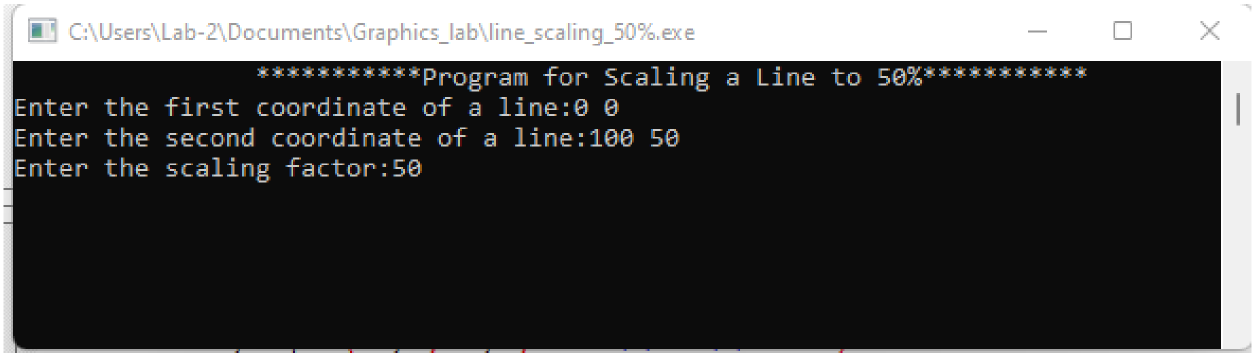
Experiment Name: Scan convert a line object from (0, 0) to (100, 50) and scale it by 50%.

Source Code:

```
#include <iostream>
#include <conio.h>
#include <graphics.h>
using namespace std;
int main()
{
   int gd=DETECT,gm;
   float x1,y1,x2,y2,sx,sy,s;
   initgraph(&gd,&gm,"C:\\Tc\\BGI");
   printf("\t\t***********Program for Scaling a Line to 50%*********** \n");
   cout<<"Enter the first coordinate of a line:";
   cin>>x1>>y1;
   cout<<"Enter the second coordinate of a line:";
   cin>>x2>>y2;
   line(x1,y1,x2,y2);
   cout<<"Enter the scaling factor:";
   cin>>s;
   sx=s/100,sy=s/100;
   setcolor(RED);
   x1=x1*sx;
```

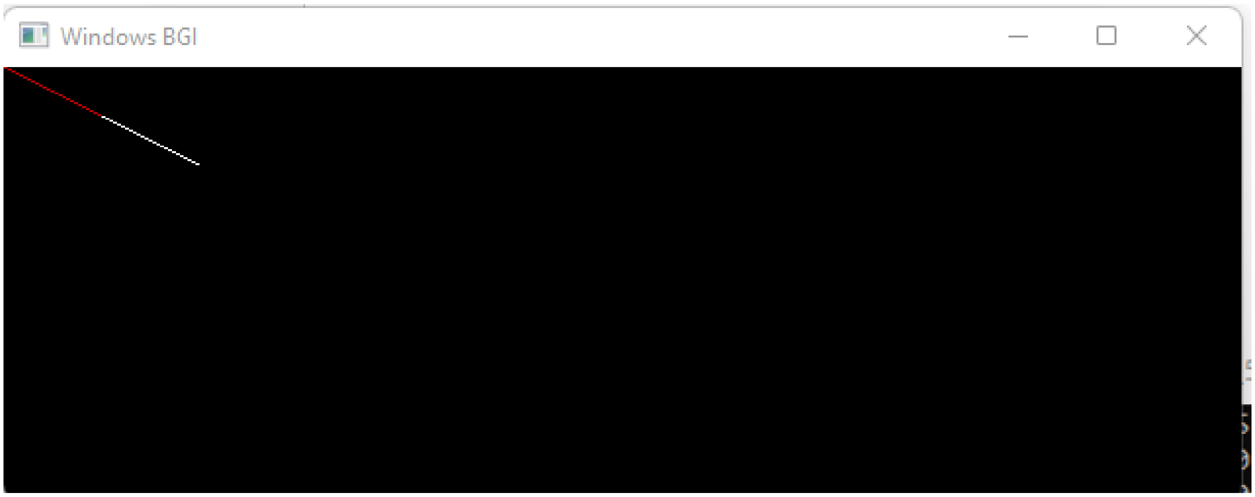
```
  y1=y1*sy;

  x2=x2*sx;

  y2=y2*sy;

  line(x1,y1,x2,y2);

  getch();

  closegraph();

}
```

Input:



```
                ***********Program for Scaling a Line to 50%***********
Enter the first coordinate of a line:0 0
Enter the second coordinate of a line:100 50
Enter the scaling factor:50
```

Output:



Experiment no: 09

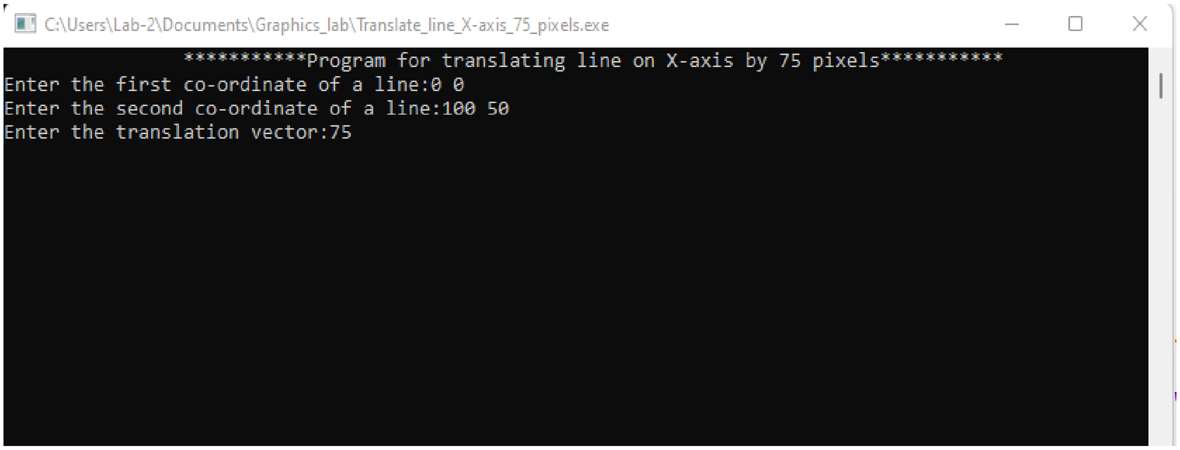
Experiment Name: Scan convert a line object from (0, 0) to (100, 50) and Translate it on X-axis by 75 pixels.

Source Code:

```
#include <iostream>
#include <conio.h>
#include <graphics.h>
using namespace std;
int main()
{
   printf("\t\t***********Program for translating line on X-axis by 75 pixels*********** \n");
   int gd=DETECT,gm,x1,x2,y1,y2,tx,ty;
   initgraph(&gd,&gm,"C:\\Tc\\BGI");
   cout<<"Enter the first co-ordinate of a line:";
   cin>>x1>>y1;
   cout<<"Enter the second co-ordinate of a line:";
   cin>>x2>>y2;
   line(x1,y1,x2,y2);
   cout<<"Enter the translation vector:";
   cin>>tx;
   setcolor(RED);
   x1=x1+tx;
   x2=x2+tx;
```

```
    line(x1,y1,x2,y2);

    getch();

    closegraph();

}
```

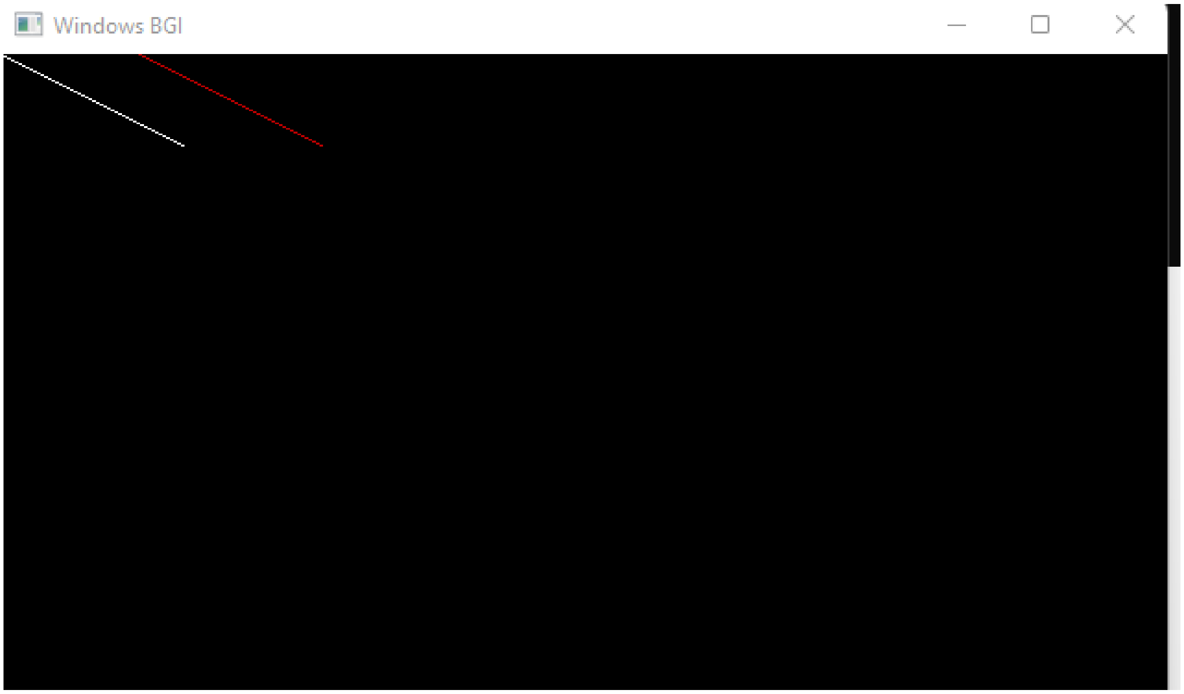
Input:



C:\Users\Lab-2\Documents\Graphics_lab\Translate_line_X-axis_75_pixels.exe

```
                ***********Program for translating line on X-axis by 75 pixels***********
Enter the first co-ordinate of a line:0 0
Enter the second co-ordinate of a line:100 50
Enter the translation vector:75
```

Output:



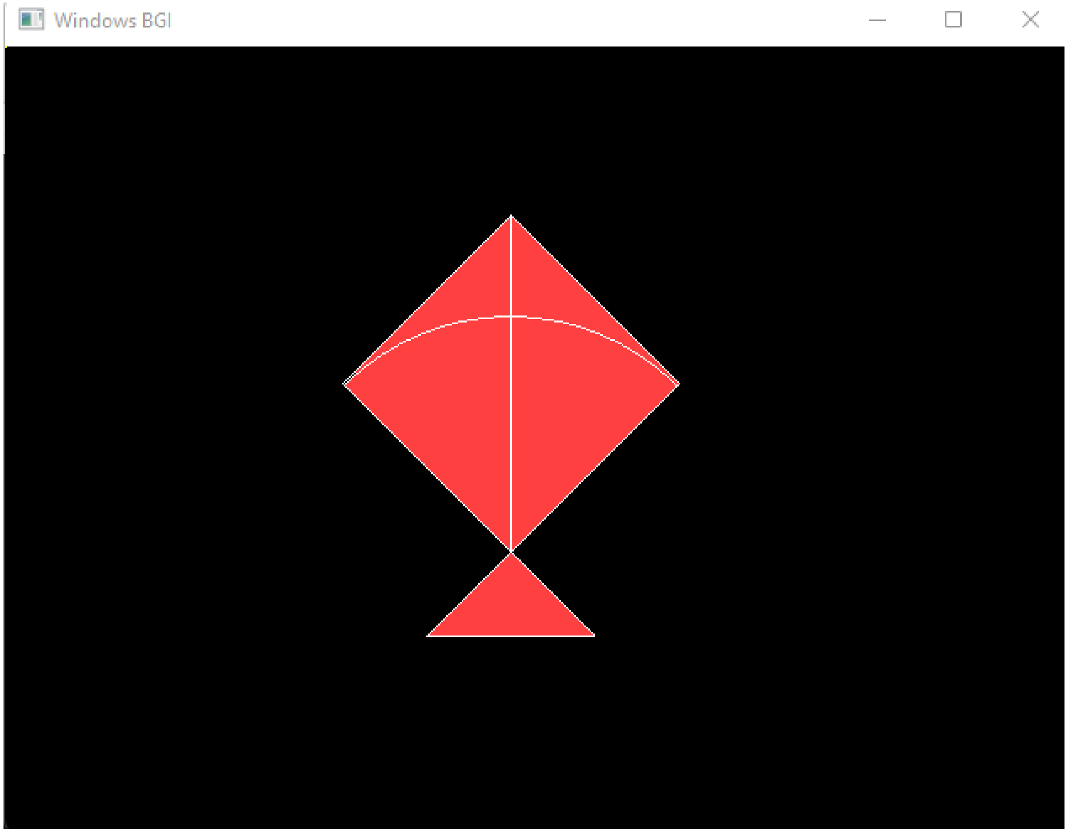
Experiment no: 10

Experiment Name: Drawing a kite using Bresenham’s line algorithm.

Source Code:

```
#include <graphics.h>
#include <iostream>
#include <conio.h>
#include <math.h>
using namespace std;
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);
    floodfill(299, 105, WHITE);
    floodfill(299, 275, WHITE);
    floodfill(301, 275, WHITE);
    line(300, 300, 250, 350);
    line(250, 350, 350, 350);
    line(300, 300, 350, 350);
    floodfill(300, 310, WHITE);
}
int main()
{
    int gd = DETECT, gm;
    initgraph(&gd, &gm, "");
    kite();
    getch();
    closegraph();
    return 0;
}
```

Output:


Windows BGI