

## Lab Report : 03



**Title: Computer Graphics Lab**  
**Course code: CSE-304**  
**3rd Year 1st Semester**

**Date of Submission: 11.06.2023**

**Submitted to-**

<p><b>Prof. Dr. Mohammad Shorif Uddin</b> <i>Professor</i> <i>Department of Computer</i> <i>Science and Engineering</i> <i>Jahangirnagar University</i> <i>Savar, Dhaka-1342</i></p>	<p><b>Dr. Morium Akther</b> <i>Associate Professor</i> <i>Department of Computer</i> <i>Science and Engineering</i> <i>Jahangirnagar University</i> <i>Savar, Dhaka-1342</i></p>
--	--

Sl	Class Roll	Registration Number	Name
01	388	20200650758	Md.Tanvir Hossain Saon

## Experiment No.07

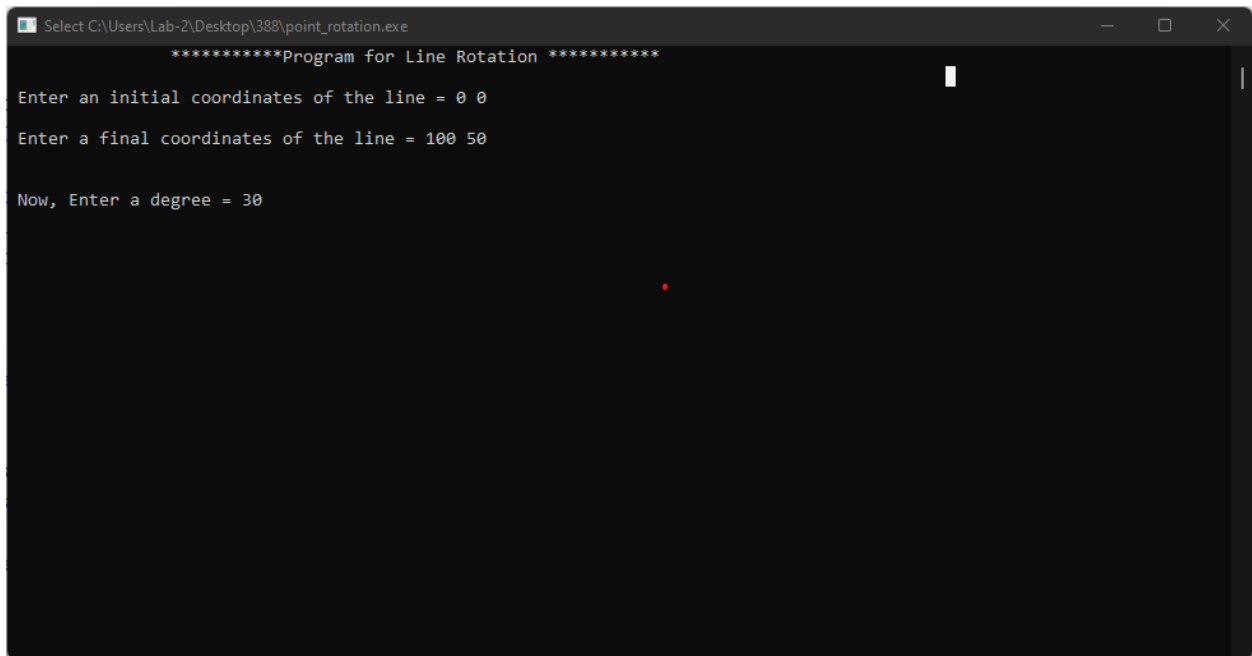
**Scan Conversion of a line object from(0,0) to (100,50):**

**i) Rotating 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();
}
```

## OUTPUT:

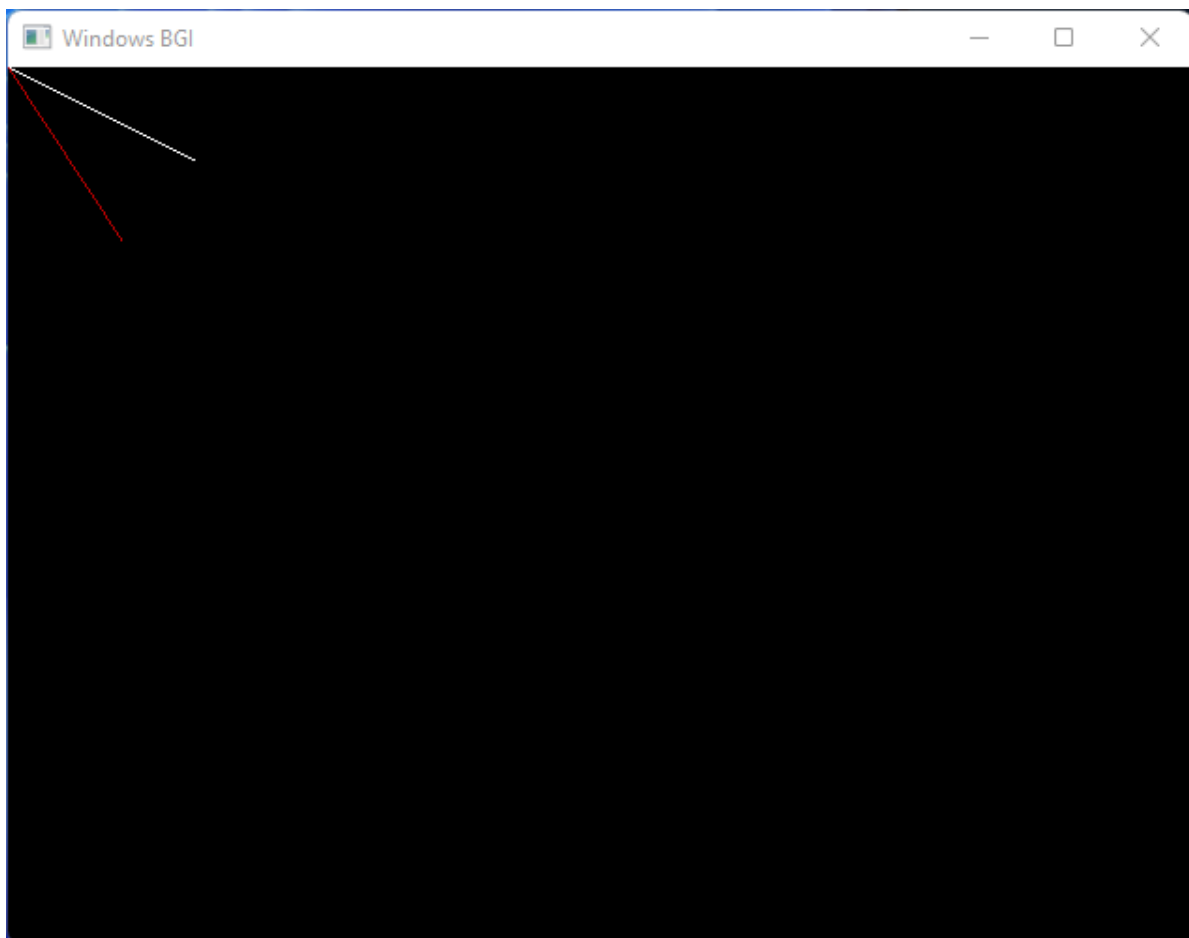


```
Select C:\Users\Lab-2\Desktop\388\point_rotation.exe

*****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
```

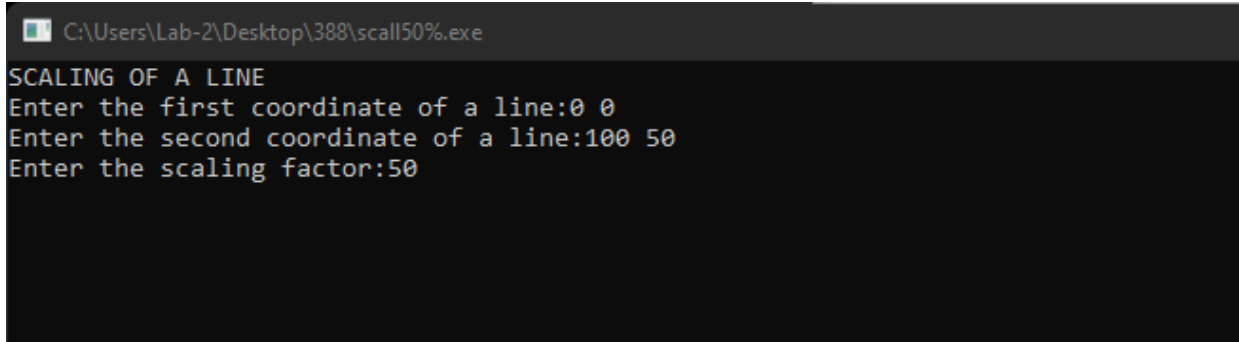


## ii) Scale it to to 50%:

### 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");
    cout<<"SCALING OF A LINE\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();
}
```

### OUTPUT:



```
C:\Users\Lab-2\Desktop\388\scall50%.exe
SCALING OF A LINE
Enter the first coordinate of a line:0 0
Enter the second coordinate of a line:100 50
Enter the scaling factor:50
```



**iii) Translate it on x axis by 75px:**

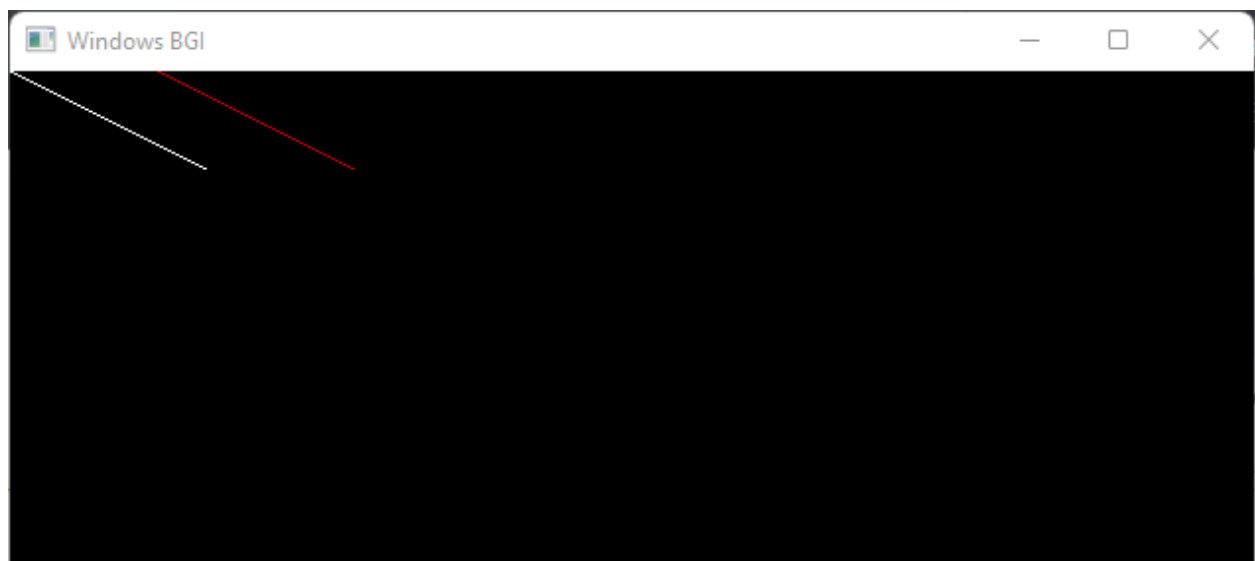
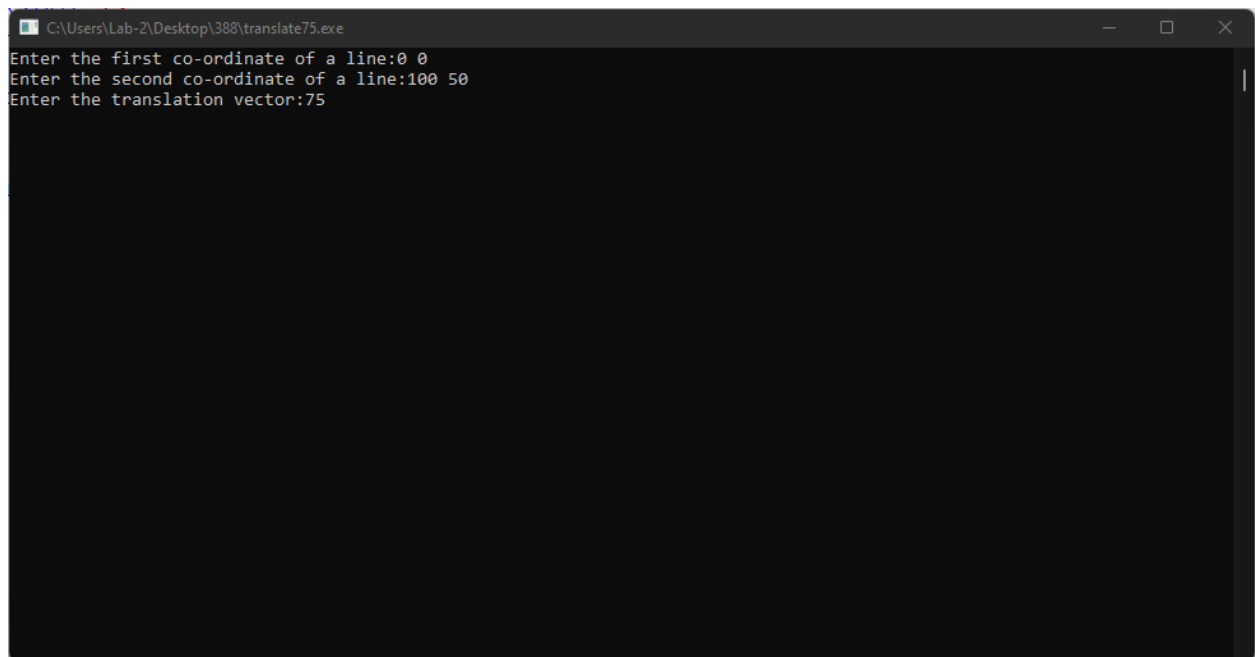
**Source Code:**

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

using namespace std;
int main()
{
    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();  
}
```

## OUTPUT:



## Experiment No.08

### Draw a kite using Bresenham's 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:

