

COMPUTER GRAPHICS AND MULTIMEDIA

LAB ASSIGNMENT 3

By: Sparsh Arya

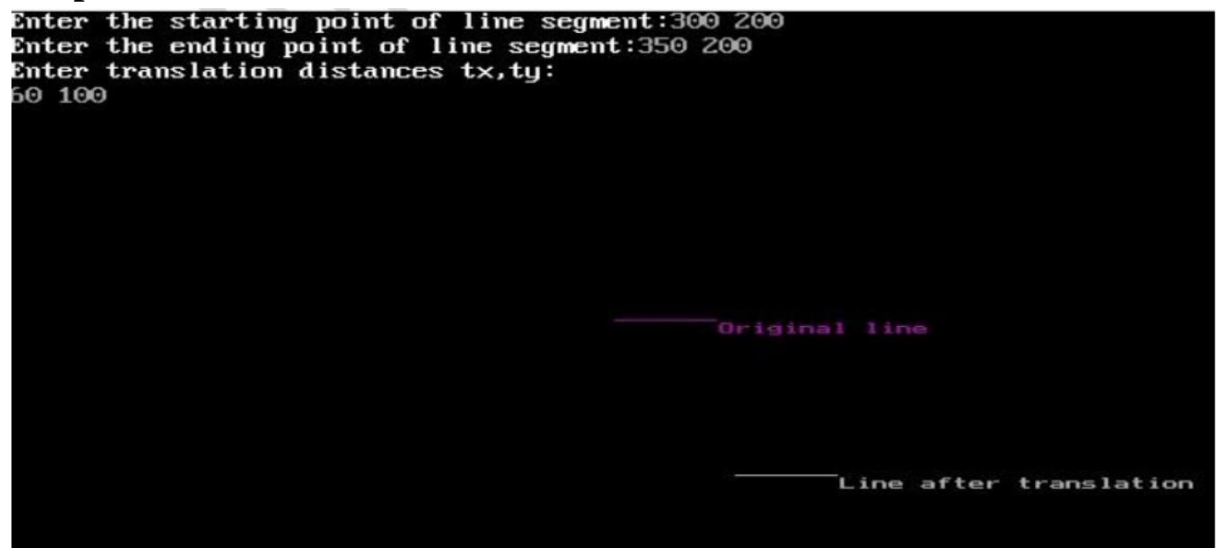
Registration number: 17BEC0656

Slot: L3+L4

Program for translation:

```
#include<stdio.h>
#include<conio.h>
#include<graphics.h>
#include<math.h>
void main()
{
    int gd=DETECT,gm;
    int x1,y1,x2,y2,tx,ty,x3,y3,x4,y4;
    initgraph(&gd,&gm,"C:\\TurboC3\\BGI");
    printf("Enter the starting point of line segment:");
    scanf("%d %d",&x1,&y1);
    printf("Enter the ending point of line segment:");
    scanf("%d %d",&x2,&y2); printf("Enter translation distances tx,ty:n");
    scanf("%d%d",&tx,&ty); setcolor(5); line(x1,y1,x2,y2); outtextxy(x2+2,y2+2,"Original
line"); x3=x1+tx;
y3=y1+ty;
x4=x2+tx;
y4=y2+ty;
setcolor(7);
line(x3,y3,x4,y4);
outtextxy(x4+2,y4+2,"Line after translation");
getch();
}
```

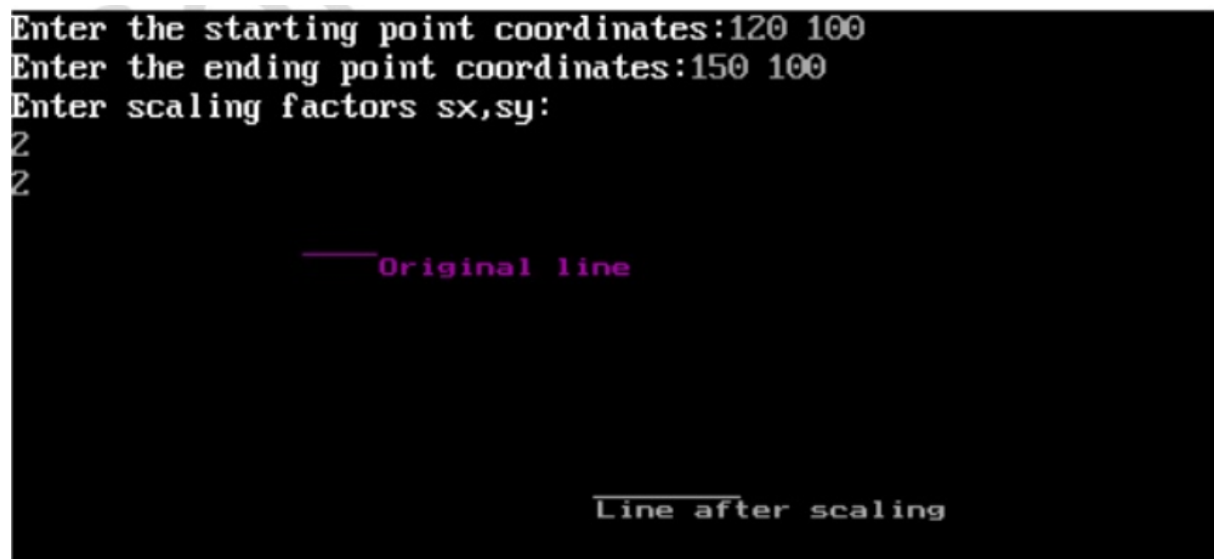
Output:



Program for scaling:

```
#include<stdio.h>
#include<conio.h>
#include<graphics.h>
#include<math.h>
void main()
{
    int gd=DETECT,gm;
    float x1,y1,x2,y2,sx,sy,x3,y3,x4,y4;
    initgraph(&gd,&gm,"C:\\TurboC3\\BGI");
    printf("Enter the starting point coordinates:");
    scanf("%f %f",&x1,&y1);
    printf("Enter the ending point coordinates:");
    scanf("%f %f",&x2,&y2);
    printf("Enter scaling factors sx,sy:n");
    scanf("%f%f",&sx,&sy); setcolor(5);
    line(x1,y1,x2,y2);
    outtextxy(x2+2,y2+2,"Original line");
    x3=x1*sx;
    y3=y1*sy;
    x4=x2*sx;
    y4=y2*sy;
    setcolor(7);
    line(x3,y3,x4,y4);
    outtextxy(x3+2,y3+2,"Line after scaling");
    getch();
}
```

Output:



```
Enter the starting point coordinates:120 100
Enter the ending point coordinates:150 100
Enter scaling factors sx,sy:
2
2

Original line


Line after scaling
```

Program for Rotation:

```
#include<stdio.h>
#include<conio.h>
#include<graphics.h>
#include<math.h>
void main()
{
    int gd=DETECT,gm;
    float x1,y1,x2,y2,x3,y3,x4,y4,a,t;
    initgraph(&gd,&gm,"C:\\TurboC3\\BGI");
    printf("Enter coordinates of starting point:n");
    scanf("%f%f",&x1,&y1);
    printf("Enter coordinates of ending pointn");
    scanf("%f%f",&x2,&y2);
    printf("Enter angle for rotationn");
    scanf("%f",&a); setcolor(5);
    line(x1,y1,x2,y2);
    outtextxy(x2+2,y2+2,"Original line");
    t=a*(3.14/180);
    x3=(x1*cos(t))-(y1*sin(t));
    y3=(x1*sin(t))+(y1*cos(t));
    x4=(x2*cos(t))-(y2*sin(t));
    y4=(x2*sin(t))+(y2*cos(t));
    setcolor(7);
    line(x3,y3,x4,y4);
    outtextxy(x3+2,y3+2,"Line after rotation");
    getch();
}
```

Output:

```
Enter coordinates of starting point:
300 200
Enter coordinates of ending point
350 200
Enter angle for rotation
45
```



The image displays a 2D coordinate system with a black background. A horizontal line segment is drawn in red, starting at the point (300, 200) and ending at (350, 200). This line is labeled "Original line" in red text to its right. Below this, a second line segment is drawn in white, starting at the same point (300, 200) and extending downwards and to the right at a 45-degree angle. This line is labeled "Line after rotation" in white text to its right.

Program for reflection along x-axis:

```
# include <stdio.h>
# include <conio.h>
# include <graphics.h>
# include <math.h>
char IncFlag;
int PolygonPoints[3][2] = {{ 10,100},{ 110,100},{ 110,200}};

void PolyLine()
{
    int iCnt;
    cleardevice();
    line(0,240,640,240);
    line(320,0,320,480);
    for (iCnt=0; iCnt<3; iCnt++)
    {
        line(PolygonPoints[iCnt][0],PolygonPoints[iCnt][1],
        PolygonPoints[(iCnt+1)%3][0],PolygonPoints[(iCnt+1)%3][1]);
    }
}

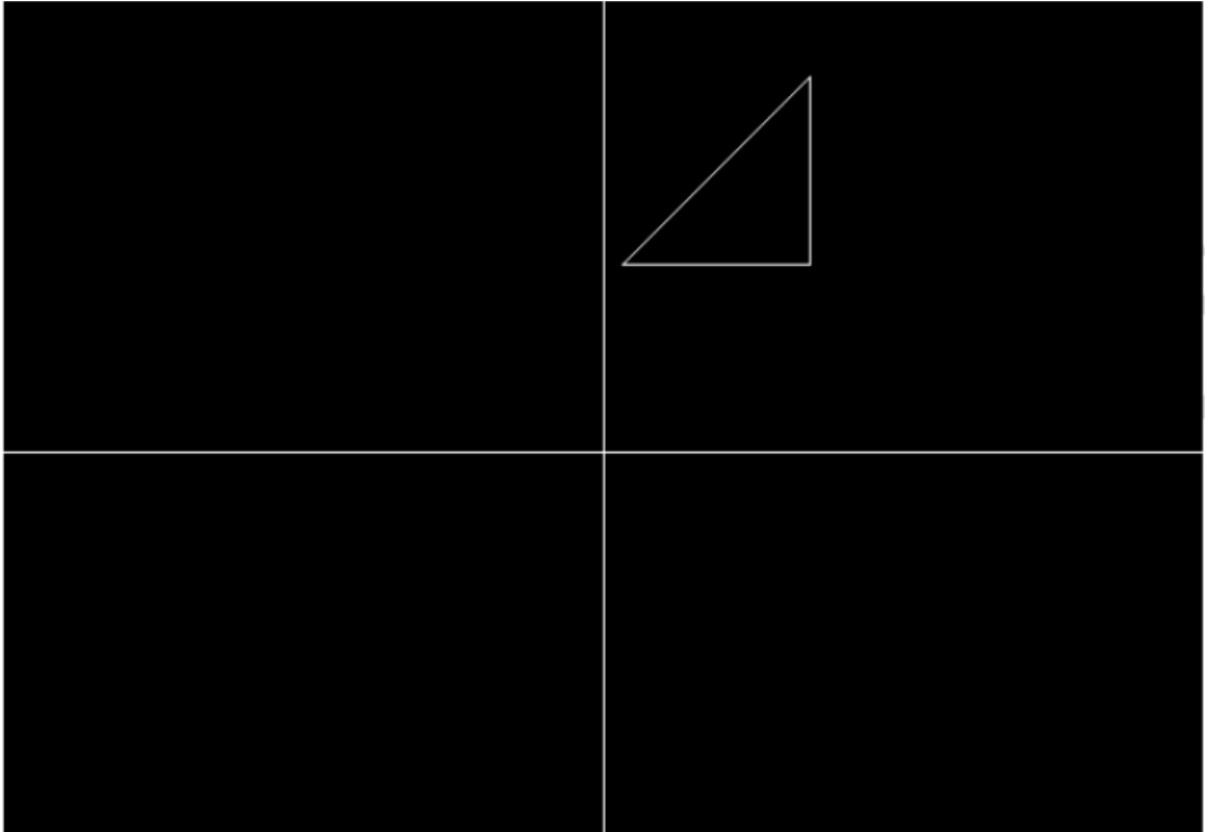
void Reflect()
{
    {
        float Angle;
        int iCnt;
        int Tx,Ty;
        printf("endl");
        for (iCnt=0; iCnt<3; iCnt++)
        {
            PolygonPoints[iCnt][1] = (480 - PolygonPoints[iCnt][1]);
        }
    }
}

void main()
{
    {
        int gDriver = DETECT, gMode;
        int iCnt;
        initgraph(&gDriver, &gMode, "C:\\TurboC3\\BGI");
        for (iCnt=0; iCnt<3; iCnt++)
        {
            PolygonPoints[iCnt][0] += 320;
            PolygonPoints[iCnt][1] = 240 - PolygonPoints[iCnt][1];
        }
        PolyLine();
        getch();
        Reflect();
    }
}
```

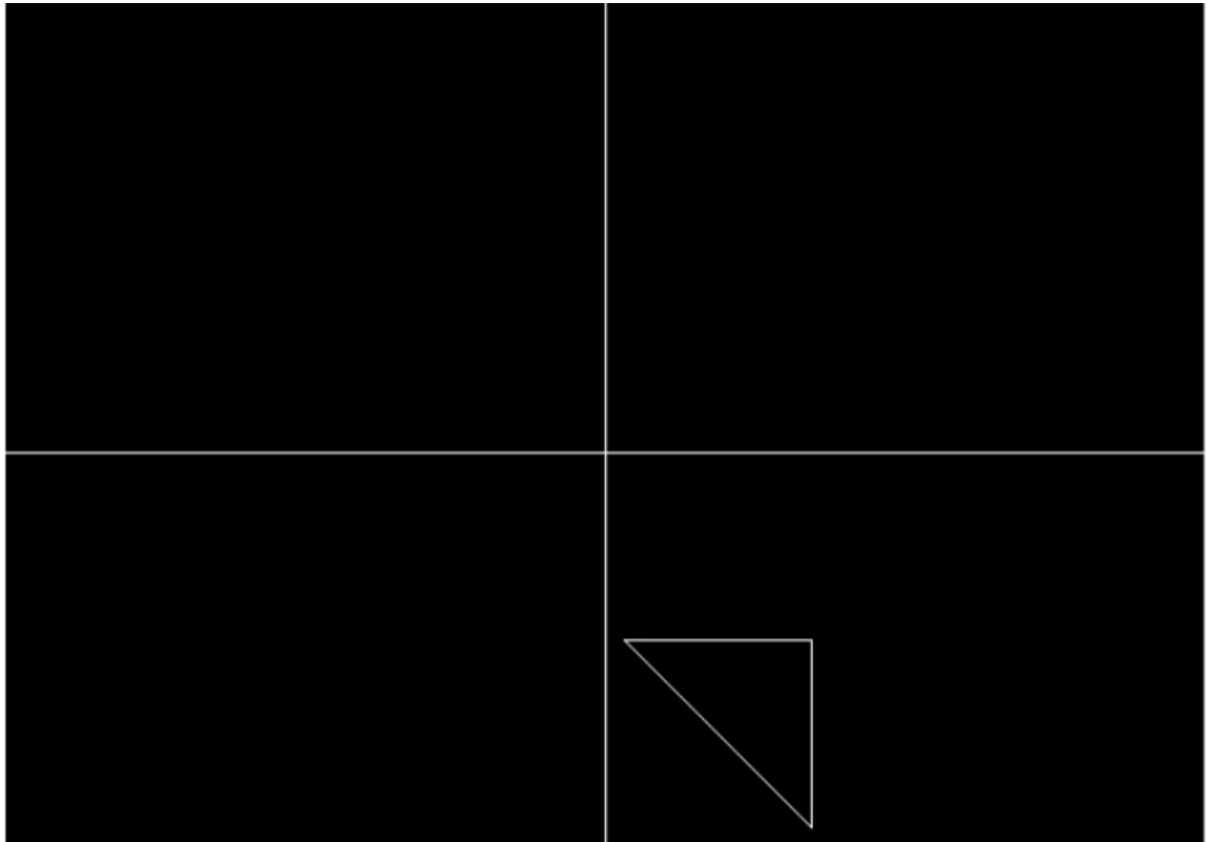
```
    PolyLine();  
    getch();  
}
```

Output:

Before reflection.



After reflection.



Program for Reflection about y-axis:

```
#include <stdio.h>

#include <conio.h>

#include <graphics.h>

#include <math.h>

char IncFlag;

int PolygonPoints[3][2] = {{10,100},{110,100},{110,200}};

void PolyLine()

{

int iCnt;

cleardevice();

line(0,240,640,240);

line(320,0,320,480);

for (iCnt=0; iCnt<3; iCnt++)

{

line(PolygonPoints[iCnt][0],PolygonPoints[iCnt][1],

PolygonPoints[(iCnt+1)%3][0],PolygonPoints[(iCnt+1)%3][1]);

}

}

void Reflect()

{

float Angle;

int iCnt;

int Tx,Ty;

for (iCnt=0; iCnt<3; iCnt++)

PolygonPoints[iCnt][0] = (640 - PolygonPoints[iCnt][0]);

}
```

```

void main()

{
int gd = DETECT, gm;
int iCnt;
initgraph(&gd, &gm, "C:\\TurboC3\\BGI");
for (iCnt=0; iCnt<3; iCnt++)
{
PolygonPoints[iCnt][0] += 320; PolygonPoints[iCnt][1] = 240 - PolygonPoints[iCnt][1];
}
PolyLine();

getch();

Reflect();

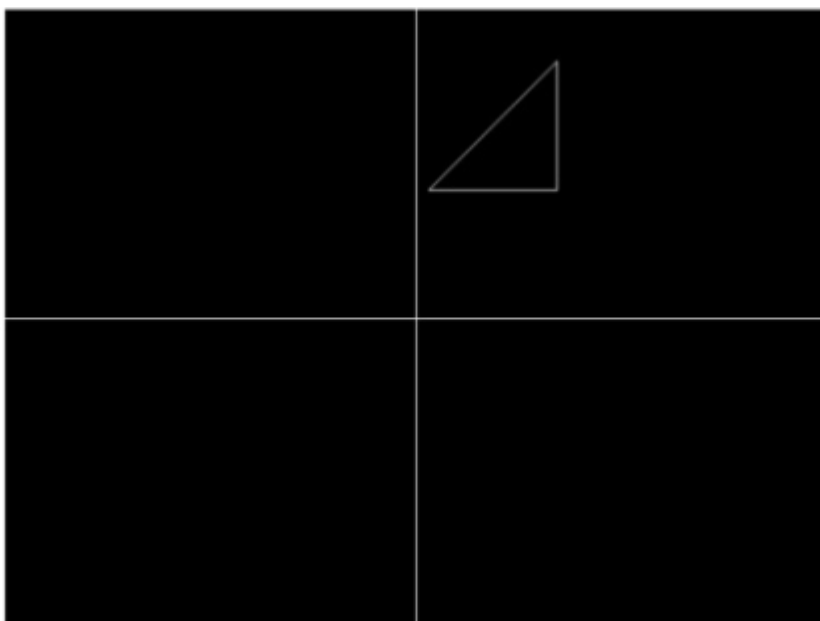
PolyLine();

getch();
}

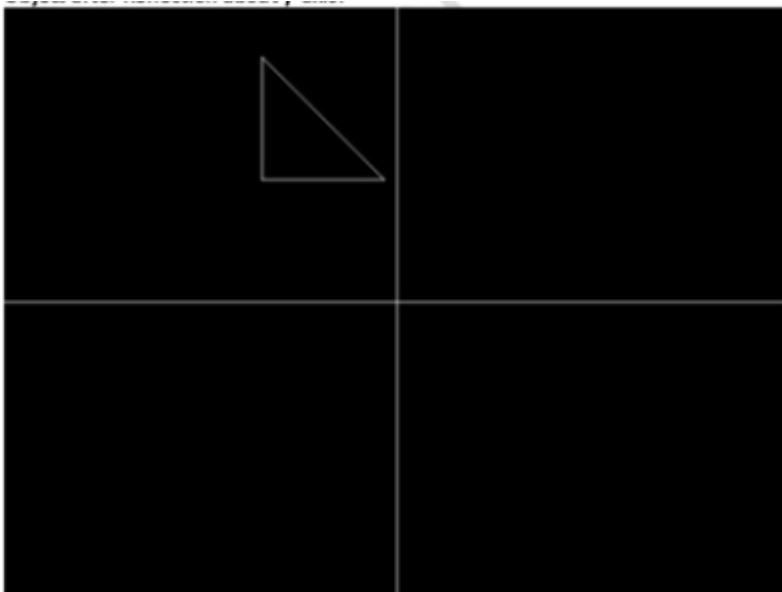
```

Output:

Before reflection.



After reflection.



Program for X-shear:

```
#include<stdio.h>

#include<conio.h>

#include<dos.h>

#include<graphics.h>

void main()

{

int gd=DETECT,gm;

float shx,shy;

initgraph(&gd,&gm,"C:\\TurboC3\\BGI");

printf("Enter shear factor shx along x-axis :");

scanf("%f",&shx);

line(100,0,200,0);

line(200,0,200,200);

line(200,200,100,200);

line(100,200,100,0);

printf("X-shear");

setcolor(12);

line((100+(0*shx)),0,(200+(0*shx)),0);

line((200+(0*shx)),0,(200+(200*shx)),200);

line((200+(200*shx)),200,(100+(200*shx)),200);

line((100+(200*shx)),200,(100+(0*shx)),0);

getch();

}
```

Output:



Program for Y-shear:

```
#include<stdio.h>

#include<conio.h>

#include<dos.h>

#include<graphics.h>

void main()

{

int gd=DETECT,gm;

float shx,shy;

initgraph(&gd,&gm,"C:\\TurboC3\\BGI");

printf("Enter shear factor shy along y-axis :");

scanf("%f",&shy);

line(100,10,200,10);

line(200,10,200,200);

line(200,200,100,200);

line(100,200,100,10);

printf("Y-shear");

setcolor(12);

line(100,10+(shy*100),200,10+(shy*200));

line(200,10+(shy*200),200,200+(shy*200));

line(200,200+(shy*200),100,200+(shy*100));

line(100,200+(shy*100),100,10+(shy*100));

getch();

closegraph();

}
```

Output:



XX End XX