Harsh Kasliwal
2003085
(2)

Page No.
Date

Experiment 6

Aim: Implement 2D Transformations on a polygon translation, Rotation, Scaling, Reflection, shearing.

Theory:-

1> Translation: To translate a point from co-ordinate position (x, y) to another (x_1, y_1) we add algebraically the translation distance T_x and T_y to original co-ordinate.

$$x_1 = x + T_x$$
$$y_1 = y + T_y$$

Matrix for Translation:
$$\begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ T_x & t_y & 1 \end{bmatrix} \text{ or } \begin{bmatrix} 1 & 0 & T_x \\ 0 & 1 & T_y \\ 0 & 0 & 1 \end{bmatrix}$$

2> Scaling: It is used to alter ~~on~~ or change the size of objects. The change is done using scaling factors. There are two scaling factors i.e S_x in x direction and S_y in y-direction.

Matrix for Scaling:
$$S = \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

3> Rotation: Process of changing the angle of the object. Rotation can be clockwise or anticlockwise. For rotation we have to specify the angle of rotation and point.

Rotation about an arbitrary point: If we want to rotate an object or point about an arbitrary point, first of all, we translate the point about which we want to rotate to the origin. Then rotate point or object about the origin, and at the end, we again translate it to the original place. We get rotation about an arbitrary point.

Matrix for homogenous co-ordinate rotation:

Clockwise

$$R = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Anticlockwise.

$$R = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

4> Reflection: It is a transformation which produces a mirror image of an object. The mirror image can be either about x or y axis.

=> Reflection about x-axis:

In this transformation value of x will remain same where the value of y will become negative. Object can be reflected about x axis with following Matrix.

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

⇒ Reflection about Y axis: Here the values of x will be reversed the value of y will remain same. Object can be reflected about Y' axis with following matrix:

$$\begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

⇒ Shearing: It is a transformation which changes the shape of the object. The sliding of layers of objects occur. The shear can be in one direction or in two directions.

⇒ Shearing in the x direction.

In this horizontal shearing sliding of layer occurs. The homogenous matrix for shearing in the x-direction is shown below.

$$\begin{bmatrix} 1 & 0 & 0 \\ Sh_x & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

⇒ Shearing in the Y direction: Here shearing is done by sliding along vertical on Y axis.

$$\begin{bmatrix} 1 & Sh_y & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

⇒ Shearing in x-y direction: Here layers will be slided in both x as well as y direction. The sliding will be in horizontal as well as vertical direction. The shape of the object will be distorted. The matrix of shear in both directions is given by.

$$\begin{bmatrix} 1 & Sh_x & 0 \\ Sh_x & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

PROGRAM:

Code:

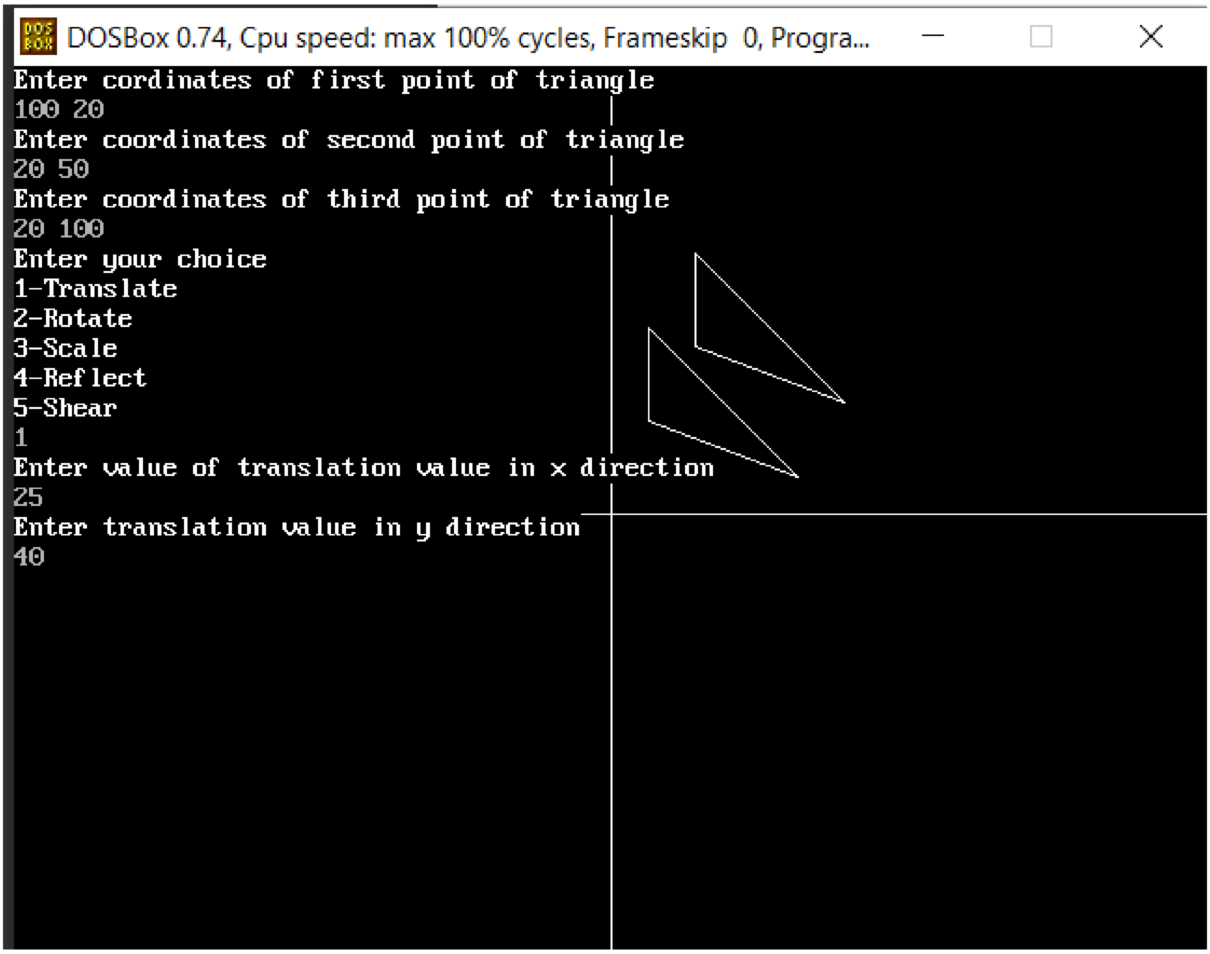
```
#include <stdio.h>
#include <conio.h>
#include <math.h>
#include <graphics.h>

void multiply(float[3][3], float[3]), clearMat(float[3][3]);
void translate(float[3], float, float);
void rotate(float[3], float);
void scale(float[3], float, float);
void reflectX(float[3]), reflectY(float[3]), reflect(float[3]);
void shear(float[3], float, float);
void Triangle(float[3], float[3], float[3]);
        void main() {
        float p1[3] = {0, 0, 1}, p2[3] = {0, 0, 1}, p3[3] = {0, 0, 1};
        float tx, ty, sx, sy, shx, shy, theta;
        int ch;
        int gd = DETECT, gm;
        initgraph(&gd, &gm, "C:\\TURBOC3\\BGI");
        line(320, 0, 320, 480); //y-axis
        line(0, 240, 640, 240); //x-axis
        printf("Enter cordinates of first point of triangle\n");
        scanf("%f %f", &p1[0], &p1[1]);
        printf("Enter coordinates of second point of triangle\n");
        scanf("%f %f", &p2[0], &p2[1]);
        printf("Enter coordinates of third point of triangle\n");
        scanf("%f %f", &p3[0], &p3[1]);
        Triangle(p1, p2, p3);
        printf("Enter your choice\n1-Translate\n2-Rotate\n3-Scale\n4-Reflect\n5-Shear\n");
        scanf("%d", &ch);
 switch(ch){
 case 1:
        printf("Enter value of translation value in x direction\n");
        scanf("%f", &tx);
        printf("Enter translation value in y direction\n");
        scanf("%f", &ty);
        translate(p1, tx, ty);
        translate(p2, tx, ty);
        translate(p3, tx, ty);
        Triangle(p1, p2, p3);
        break;
 case 2:
        printf("Enter angle of rotation in degrees\n");
        scanf("%f", &theta);
        theta *= 3.1415f/180;
        rotate(p1, theta);
        rotate(p2, theta);
        rotate(p3, theta);
        Triangle(p1, p2, p3);
        break;
 case 3:
        printf("Enter scale value in x direction\n");
        scanf("%f", &sx);
        printf("Enter scale value in y direction\n");
        scanf("%f", &sy);
        scale(p1, sx, sy);
        scale(p2, sx, sy);
        scale(p3, sx, sy);
        Triangle(p1, p2, p3);
        break;
case 4:
        printf("Reflection about what ?\n");
        printf("1 - X axis\n2- Y axis\n3 - Origin");
        scanf("%d", &ch);
        if(ch == 1){
        reflectX(p1);
        reflectX(p2);
        reflectX(p3);
        }
        else if(ch == 2){
        reflectY(p1);
        reflectY(p2);
        reflectY(p3);
        }
        else{
        reflect(p1);
        reflect(p2);
        reflect(p3);
        }
        Triangle(p1, p2, p3);
        break;
case 5:
        printf("Enter shearing value in x direction\n");
        scanf("%f", &shx);
```

```
            printf("Enter shearing value in y direction\n");
            scanf("%f", &shy);
            shear(p1, shx, shy);
            shear(p2, shx, shy);
            shear(p3, shx, shy);
            Triangle(p1, p2, p3);
            break;
 }
 getch();
 closegraph();
}
void translate(float p[], float tx, float ty){
            int i, j;
            float mat[3][3];
            clearMat(mat);
 mat[0][0] = 1;
 mat[1][1] = 1;
 mat[2][2] = 1;
 mat[0][2] = tx;
 mat[1][2] = ty;
 multiply(mat, p);
}
void rotate(float p[], float theta){
            int i, j;
            float mat[3][3];
            clearMat(mat);
 mat[0][0] = cos(theta);
 mat[0][1] = -sin(theta);
 mat[1][0] = sin(theta);
 mat[1][1] = cos(theta);
 mat[2][2] = 1;
 multiply(mat, p);
}
void scale(float p[3], float sx, float sy){
             int i, j;
            float mat[3][3];
            clearMat(mat);
 mat[0][0] = sx;
 mat[1][1] = sy;
 mat[2][2] = 1;
 multiply(mat, p);
}
void reflectX(float p[3]){
            float mat[3][3];
            clearMat(mat);
 mat[0][0] = 1;
 mat[1][1] = -1;
 mat[2][2] = 1;
 multiply(mat, p);
}
void reflectY(float p[3]){
            float mat[3][3];
            clearMat(mat);
 mat[0][0] = -1;
 mat[1][1] = 1;
 mat[2][2] = 1;
 multiply(mat, p);
}
```

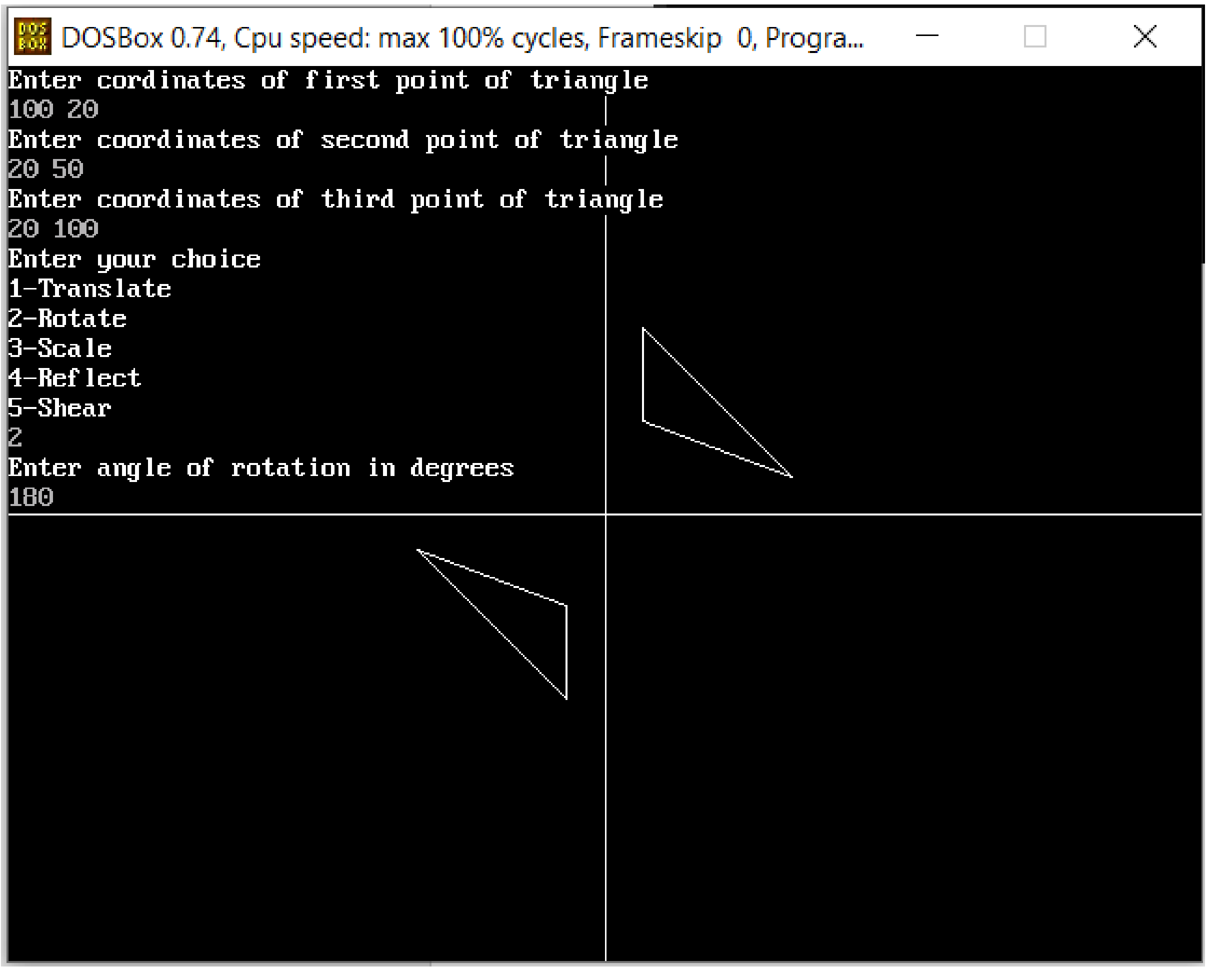
```
void reflect(float p[3]){
            float mat[3][3];
            clearMat(mat);
 mat[0][0] = -1;
 mat[1][1] = -1;
 mat[2][2] = 1;
 multiply(mat, p);
}
void shear(float p[3], float shx, float shy){
            float mat[3][3];
            clearMat(mat);
 mat[0][0] = 1; mat[1][1] = 1; mat[2][2] = 1;
 mat[1][0] = shx;
 mat[0][1] = shy;
 multiply(mat, p);
}
void multiply(float mat[3][3], float pt[3]){
            int pt0 = pt[0], pt1 = pt[1], pt2 = pt[2];
 pt[0] = pt0*mat[0][0] + pt1*mat[0][1] + pt2*mat[0][2];
 pt[1] = pt0*mat[1][0] + pt1*mat[1][1] + pt2*mat[1][2];
 pt[2] = pt0*mat[2][0] + pt1*mat[2][1] + pt2*mat[2][2];
}
void Triangle(float p1[], float p2[], float p3[]){
 line(320+p1[0], 240-p1[1], 320+p2[0], 240-p2[1]);
 line(320+p2[0], 240-p2[1], 320+p3[0], 240-p3[1]);
 line(320+p3[0], 240-p3[1], 320+p1[0], 240-p1[1]);
}
void clearMat(float mat[3][3]){
 int i, j;
for(i = 0; i < 3; i++)
        for(j = 0; j < 3; j++)
                mat[i][j] = 0;
}
```

OUTPUT:



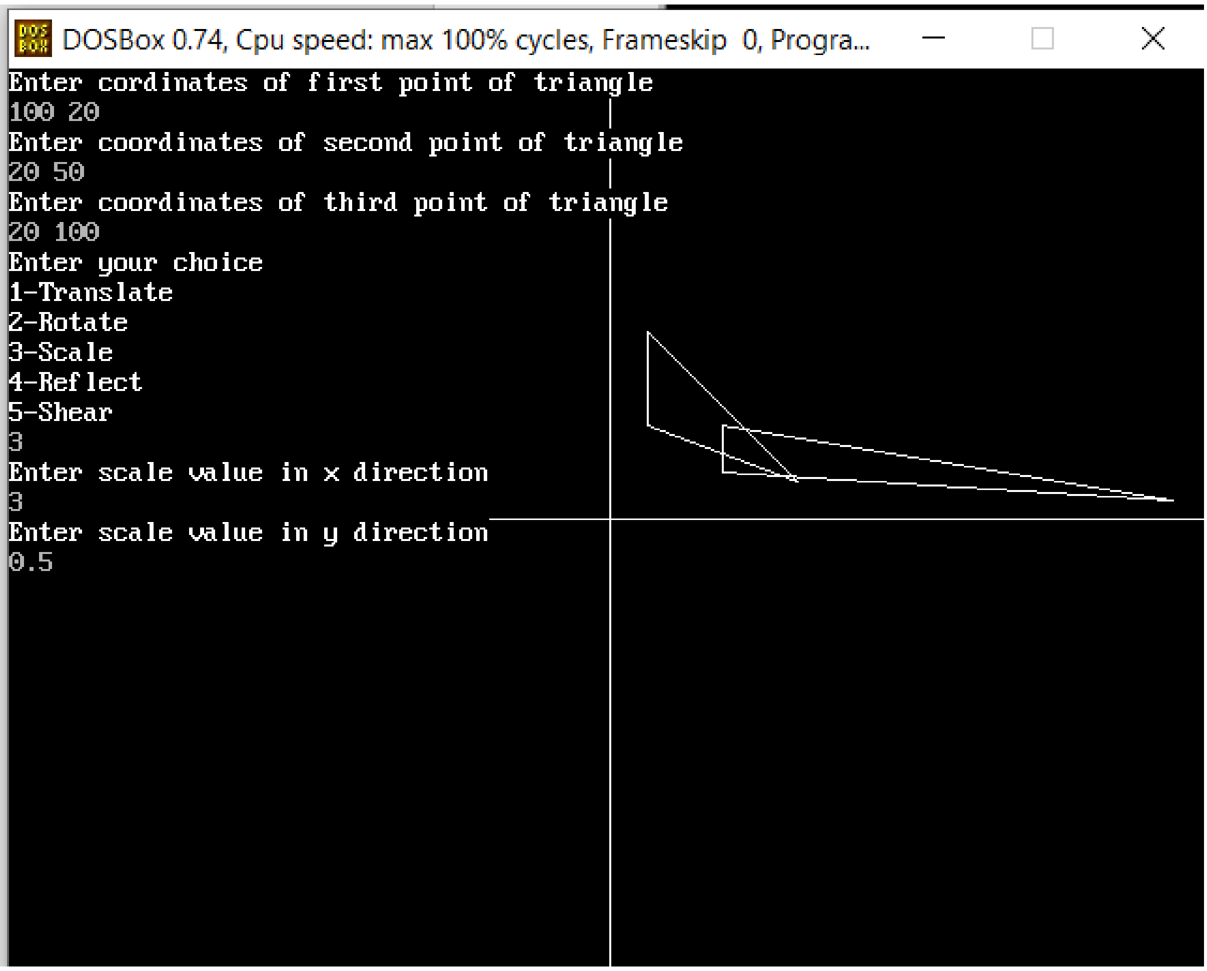
DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Progra...

```
Enter cordinates of first point of triangle
100 20
Enter coordinates of second point of triangle
20 50
Enter coordinates of third point of triangle
20 100
Enter your choice
1-Translate
2-Rotate
3-Scale
4-Reflect
5-Shear
1
Enter value of translation value in x direction
25
Enter translation value in y direction
40
```



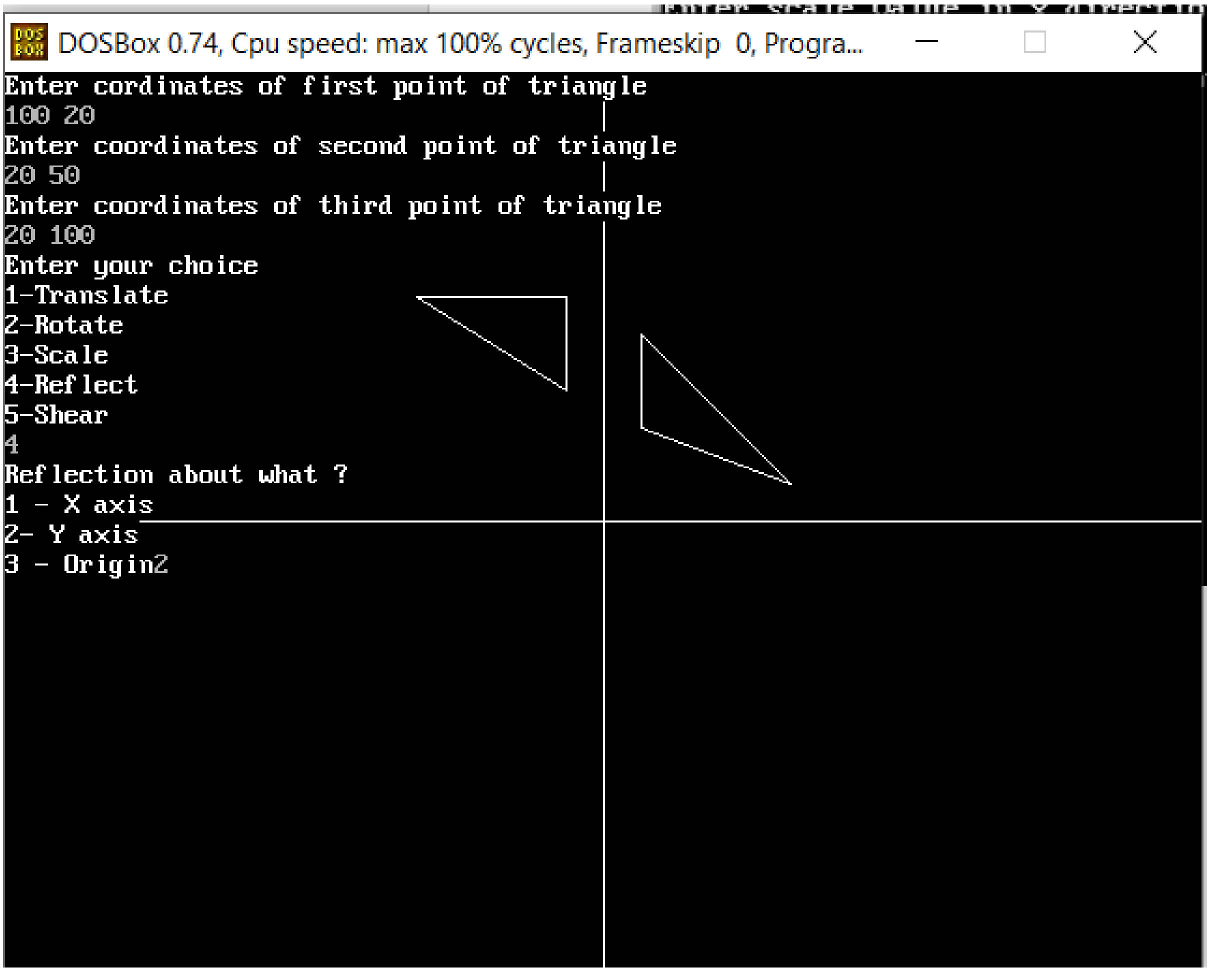
DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Progra...

```
Enter cordinates of first point of triangle
100 20
Enter coordinates of second point of triangle
20 50
Enter coordinates of third point of triangle
20 100
Enter your choice
1-Translate
2-Rotate
3-Scale
4-Reflect
5-Shear
2
Enter angle of rotation in degrees
180
```



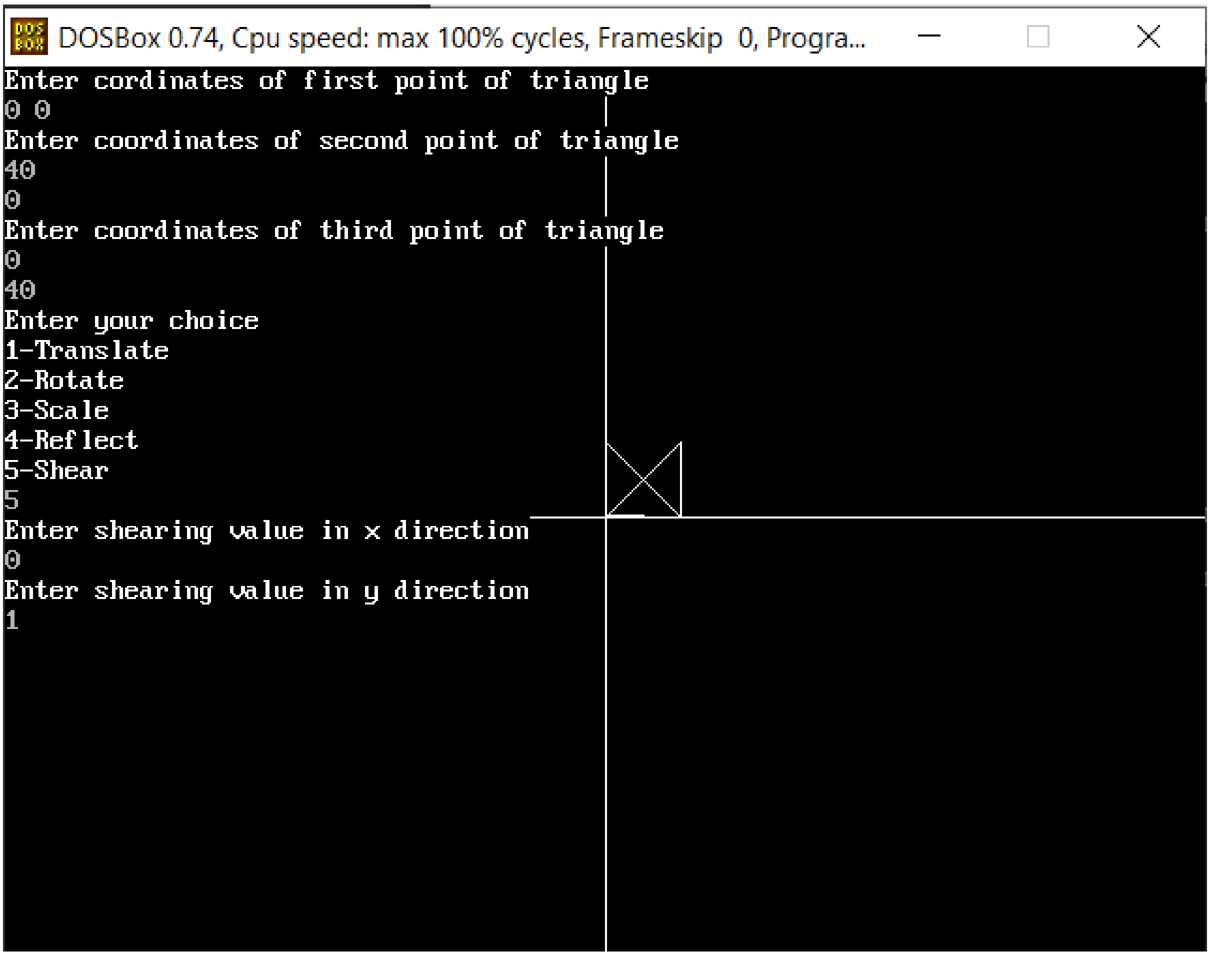
DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Progra...

```
Enter cordinates of first point of triangle
100 20
Enter coordinates of second point of triangle
20 50
Enter coordinates of third point of triangle
20 100
Enter your choice
1-Translate
2-Rotate
3-Scale
4-Reflect
5-Shear
3
Enter scale value in x direction
3
Enter scale value in y direction
0.5
```



DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Progra...

```
Enter cordinates of first point of triangle
100 20
Enter coordinates of second point of triangle
20 50
Enter coordinates of third point of triangle
20 100
Enter your choice
1-Translate
2-Rotate
3-Scale
4-Reflect
5-Shear
4
Reflection about what ?
1 - X axis
2- Y axis
3 - Origin2
```



DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Progra...

```
Enter cordinates of first point of triangle
0 0
Enter coordinates of second point of triangle
40
0
Enter coordinates of third point of triangle
0
40
Enter your choice
1-Translate
2-Rotate
3-Scale
4-Reflect
5-Shear
5
Enter shearing value in x direction
0
Enter shearing value in y direction
1
```