Computer Graphics Lab Experiment No: - 08
Aim: Implement 2D Transformations: Translation, Scaling, Rotation, Reflection, Shear.

2D Translation

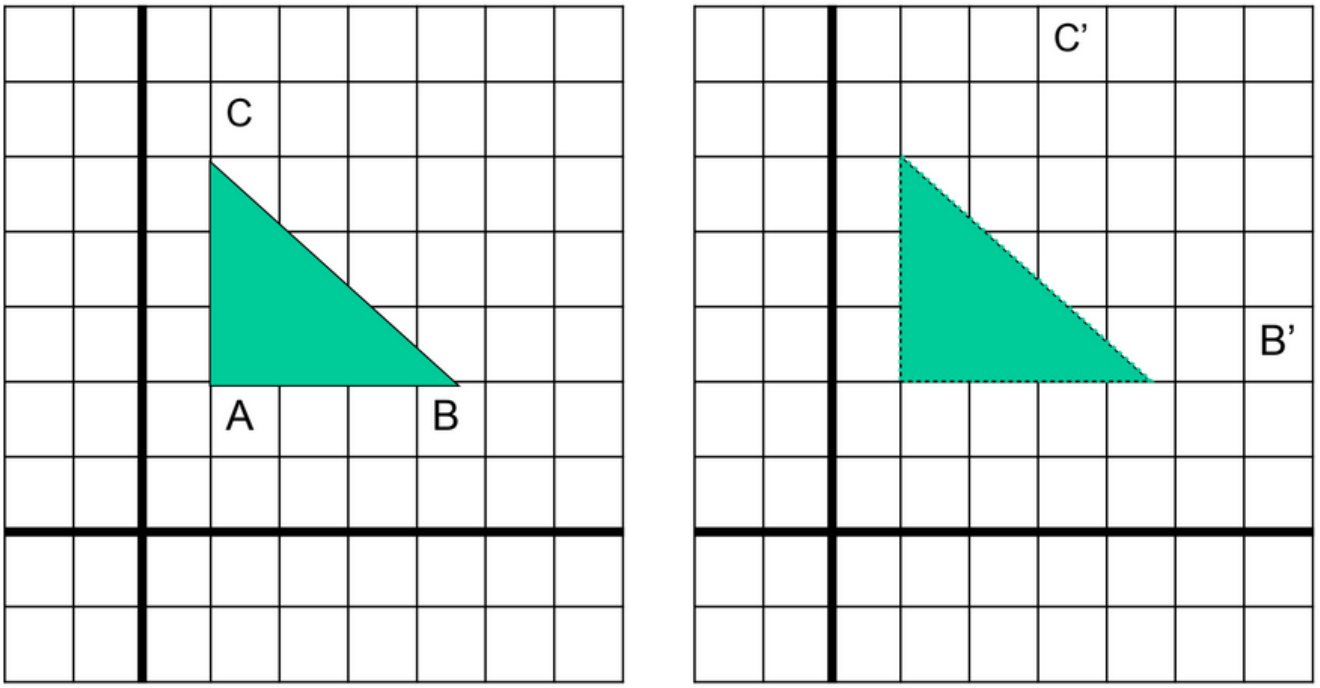
A translation is a displacement in a particular direction.

Repositioning an object along a straight line path from one co-ordinate location to another

(x,y) —---> (x’,y’)

To translate a 2D position, we add translation distances tx and ty to the original coordinates (x,y) to obtain the new coordinate position (x’,y’)

x’= x + tx , y’= y + ty



Scaling

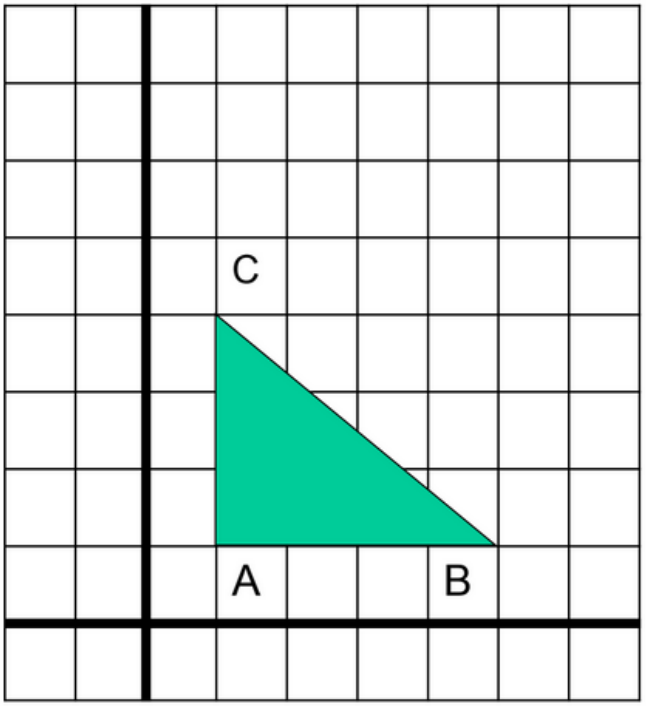
• Scaling changes the size of an object and involves two scale factors, Sx and Sy for the x- and y- coordinates respectively.

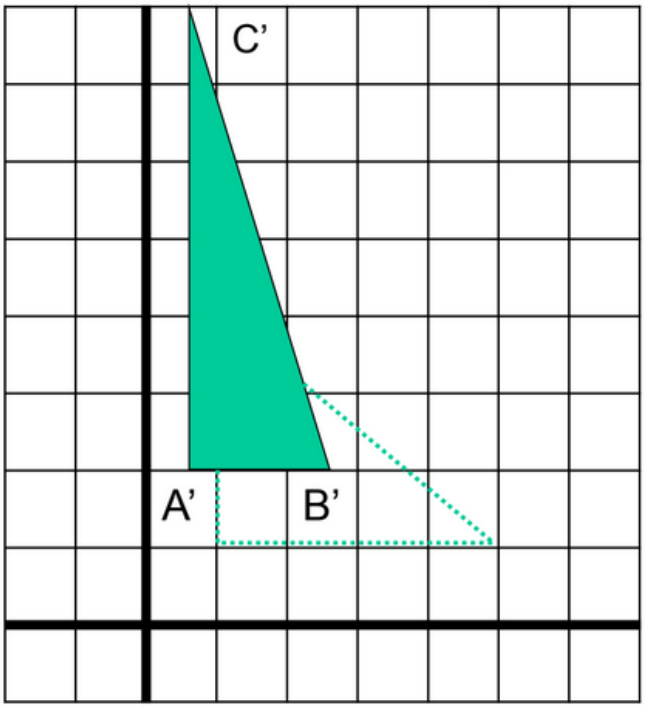
• Scales are about the origin.

• We can write the components:

p'x = sx*px

p'y = sy*py





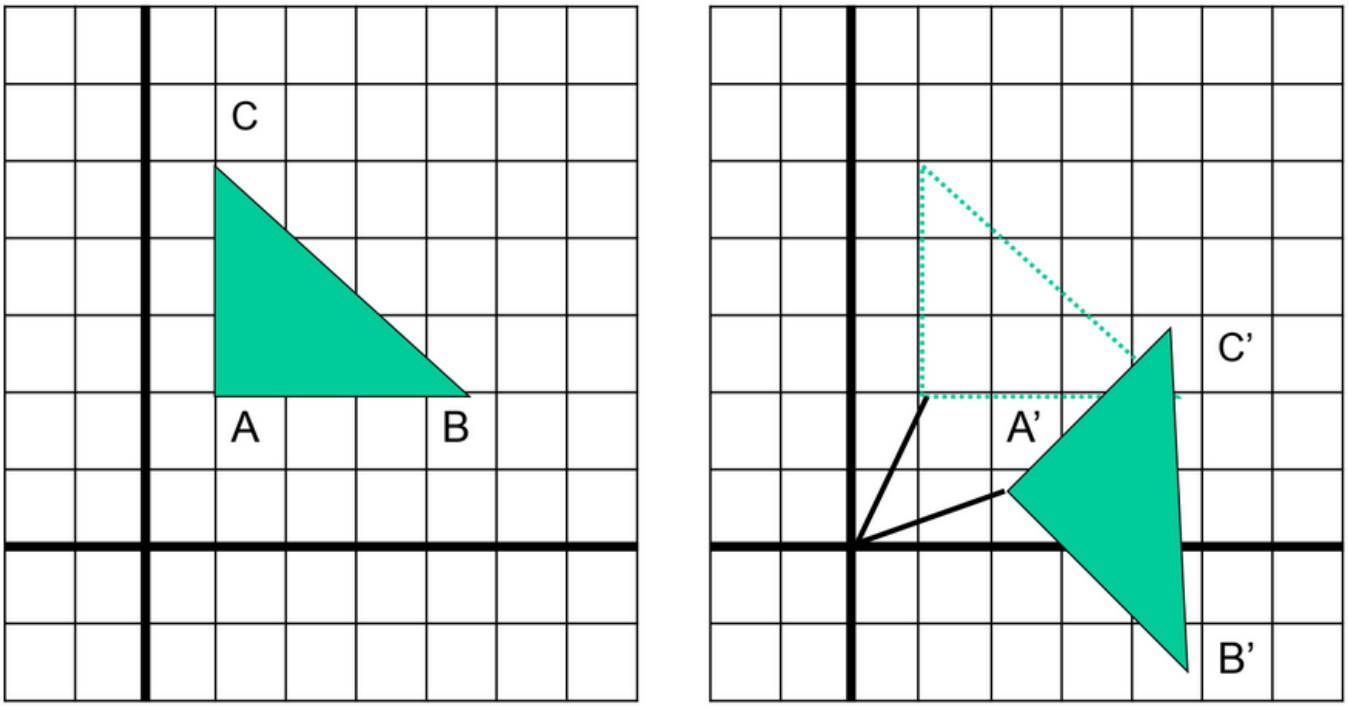
Rotation

• A rotation repositions all points in an object along a circular path in the plane centered at the pivot point.

• First, we'll assume the pivot is at the origin.
We can write the components:

$p'x = px \cos\theta - py \sin\theta$ $p'y = px \sin\theta + py \cos\theta$

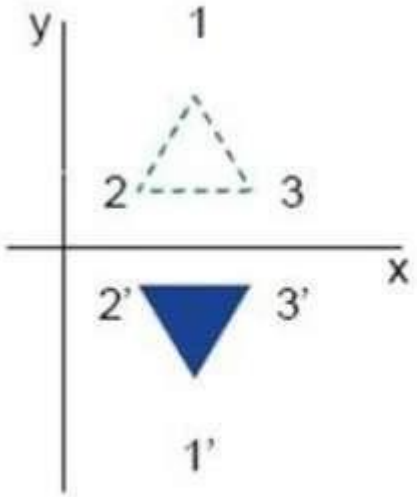
θ can be clockwise (-ve) or counterclockwise (+ve as our example).



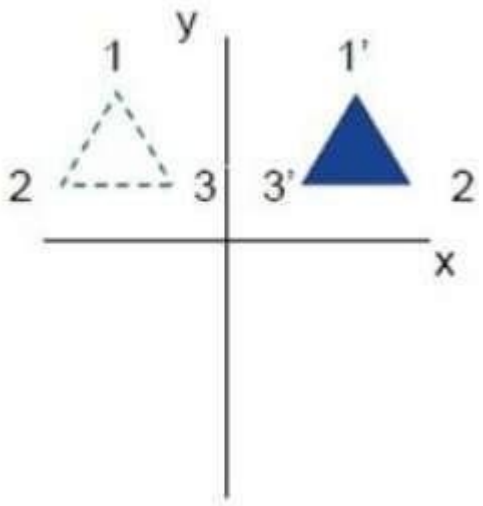
Reflections

• A reflection is a reversal of an object with respect to a line in 2 dimensions or a plane in 3 dimensions.

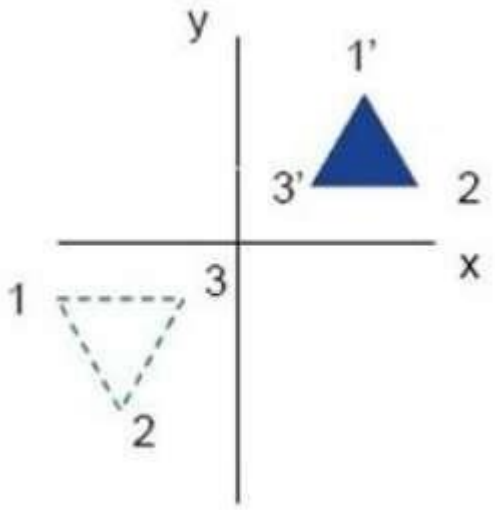
• Generally we reflect in a line or plane through the origin.



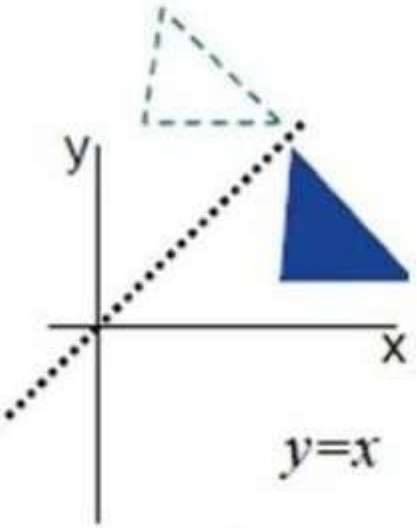
(a)



(b)



(c)



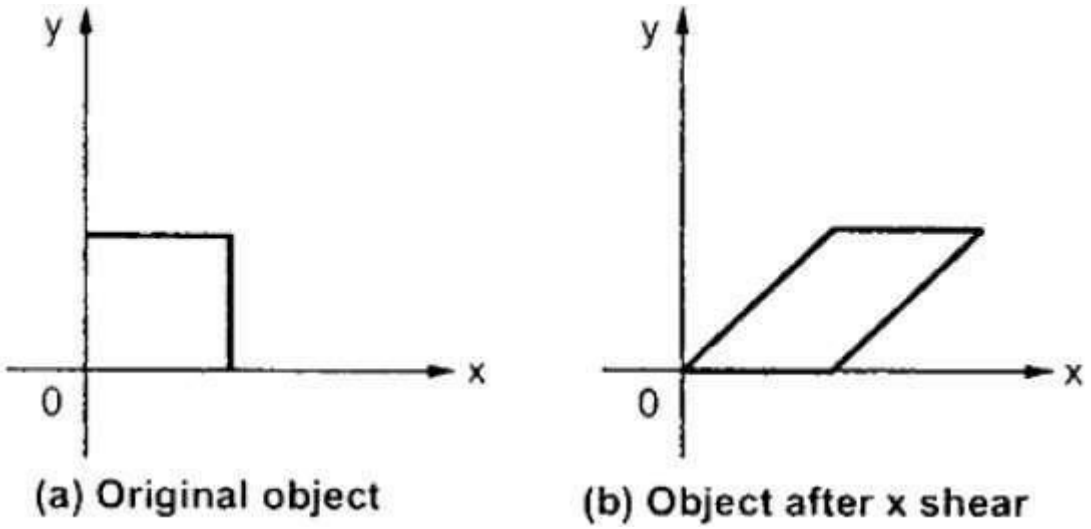
(d)

Shear

A transformation that slants the shape of an object is called the shear transformation. There are two shear transformations X-Shear and Y-Shear. One shifts X coordinates values and other shifts Y coordinate values. However; in both the cases only one coordinate changes its coordinates and other preserves its values. Shearing is also termed as Skewing.

X-Shear

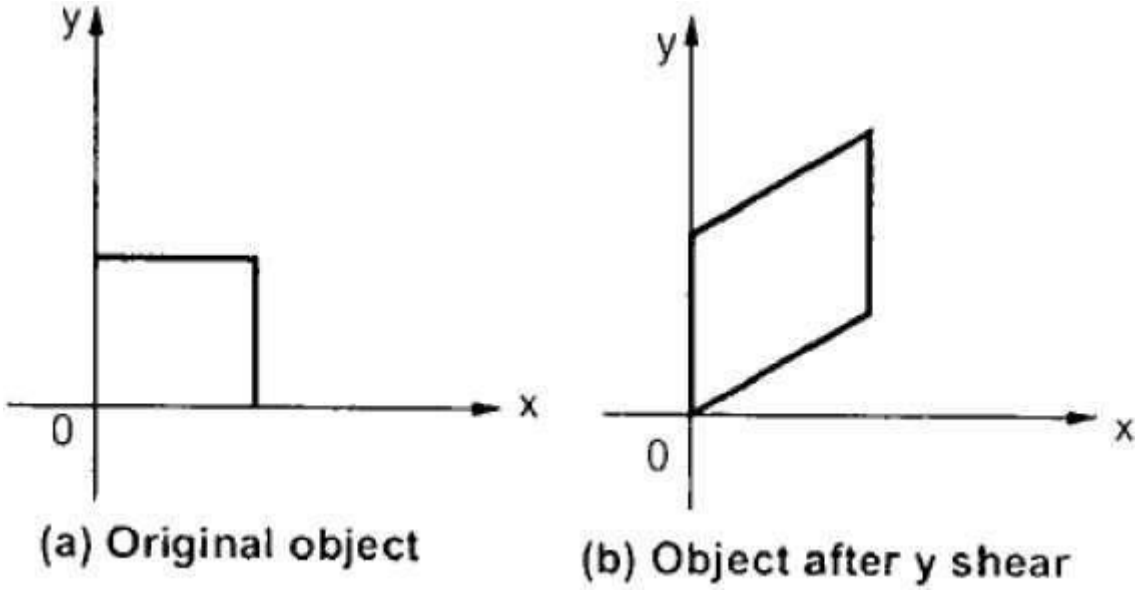
The X-Shear preserves the Y coordinate and changes are made to X coordinates, which causes the vertical lines to tilt right or left as shown in below figure.



(a) Original object (b) Object after x shear

Y-Shear

The Y-Shear preserves the X coordinates and changes the Y coordinates which causes the horizontal lines to transform into lines which slopes up or down as shown in the following figure.



Code

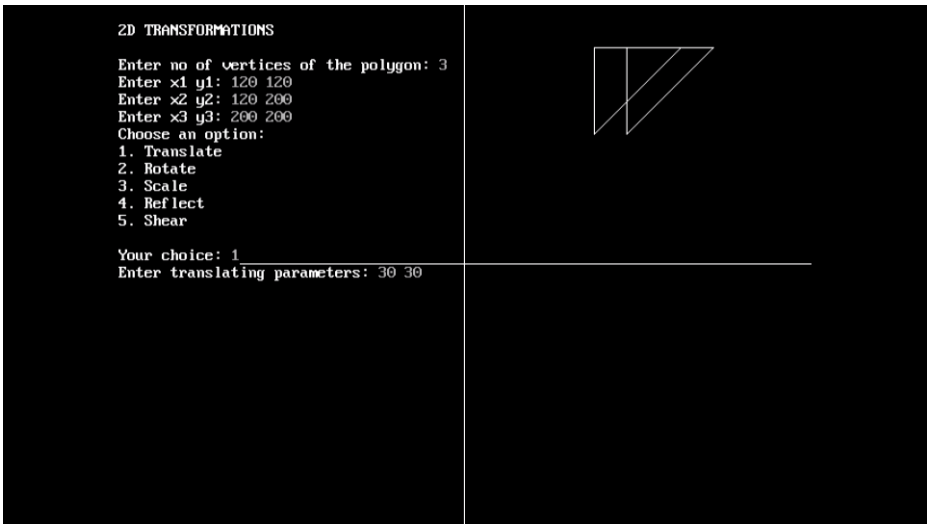
```
#include <stdio.h>
#include <math.h>
#include <graphics.h>
#define SIN(x) sin(3.141592653589793 * x / 180)
#define COS(x) cos(3.141592653589793 * x / 180)
void translate(int x[], int y[], int tx, int ty, int n) {
        int i;
        for(i = 0; i < n; i++) {
   x[i] += tx;
   y[i] += ty;
        }
}
void rotate(int x[], int y[], double angle, int n) {
        int i, X, Y;
```

```
        for(i = 0; i < n; i++)
   X = x[i], Y = y[i];
   x[i] = X * COS(angle) - Y * SIN(angle);
   y[i] = X * SIN(angle) + Y * COS(angle);
}
void scale(int x[], int y[], float sx, float sy, int n) {
        int i;
        for(i = 0; i < n; i++)
   x[i] *= sx;
   y[i] *= sy;
}

void reflect(int x[], int y[], int axis, int n) {
        int i;
        for(i = 0; i < n; i++) {
   if(axis == 0)
        y[i] *= -1;
   else
        x[i] *= -1;
}
}

void shear(int x[], int y[], float shx, float shy, int n)
{
        int i, X, Y;
        for(i = 0; i < n; i++)
   X = x[i], Y = y[i];
   x[i] += shx * Y;
```

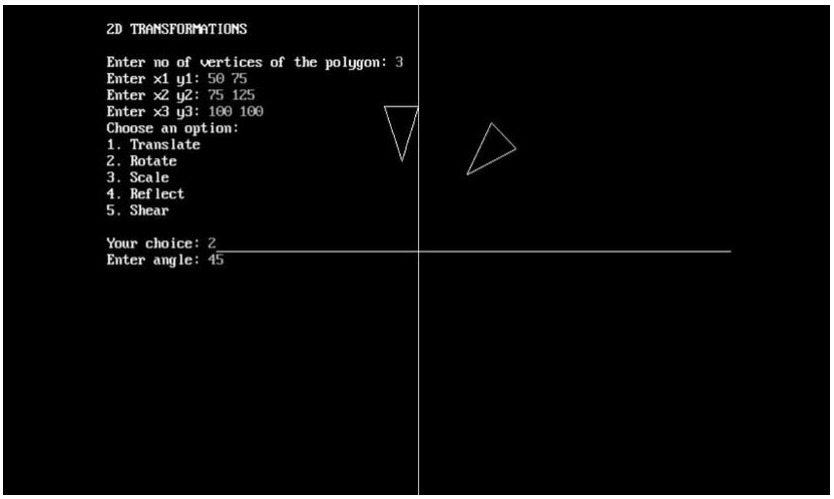
```
    y[i] += shy * X;
};
int main()
{
        int i, n, xc, yc, gd = DETECT, gm, choice, tx, ty, angle, axis, x[100], y[100],
input[200];
        float sx, sy, shx, shy;
        initgraph(&gd, &gm, "C:\\TURBOC3\\BGI");
        xc = getmaxx() / 2;
        yc = getmaxy() / 2;
        printf("\n2D TRANSFORMATIONS\n\nEnter no of vertices of the polygon: ");
        scanf("%d", &n);
    for(i = 0; i < n; i++)
         {printf("Enter x%d y%d: ", i+1, i+1);
         scanf("%d %d", &x[i], &y[i]);
         input[2 * i] = xc + x[i];
         input[2 * i + 1] = yc - y[i];
    }
    input[2 * n] = input[0];
    input[2 * n + 1] = input[1];
    line(0, yc, getmaxx(), yc);
    line(xc, 0, xc, getmaxy());
    drawpoly(n + 1, input);
    printf("Choose an option:\n1. Translate\n2. Rotate\n3. Scale\n4. Reflect\n5. Shear\n\nYour
choice: ");
    scanf("%d", &choice);
    switch(choice) {
         case 1:
```

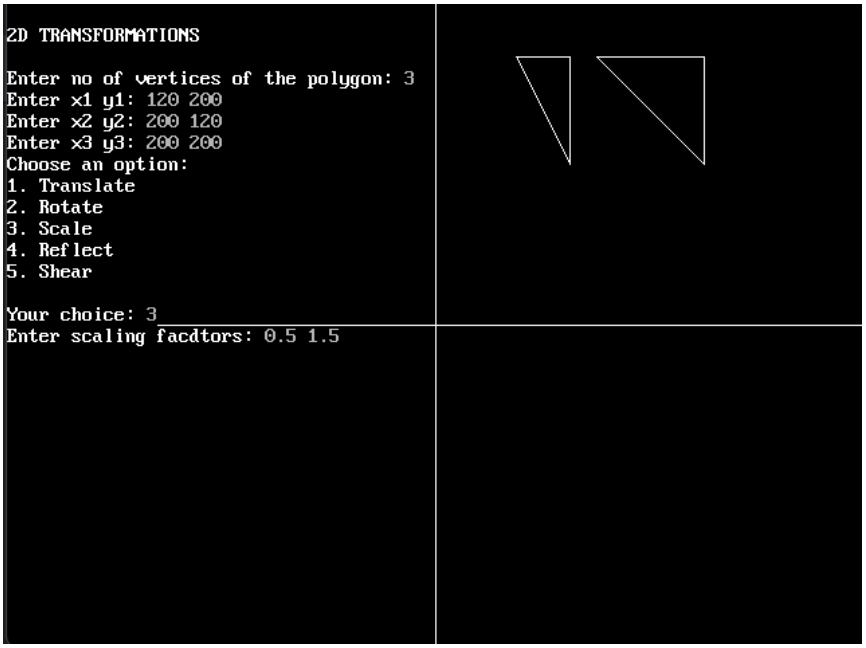
```
            printf("Enter translating parameters: ");
            scanf("%d %d", &tx, &ty);
            translate(x, y, tx, ty, n);
            break;
        case 2:
        printf("Enter angle: ");
        scanf("%d", &angle);
        rotate(x, y, angle, n);
        break;
    case 3:
        printf("Enter scaling facdtors: ");
        scanf("%f %f", &sx, &sy);
        scale(x, y, sx, sy, n);
        break;
    case 4:
        printf("Enter 0 to reflect about x-axis and 1 to reflect about y-axis: ");
        scanf("%d", &axis);
        reflect(x, y, axis, n);
        break;
    case 5:
        printf("Enter shearing factors: ");
        scanf("%f %f", &shx, &shy);
        shear(x, y, shx, shy, n);
        break;
    default:
        printf("Invalid choice\n");
        }
        for(i = 0; i < n; i++)
```

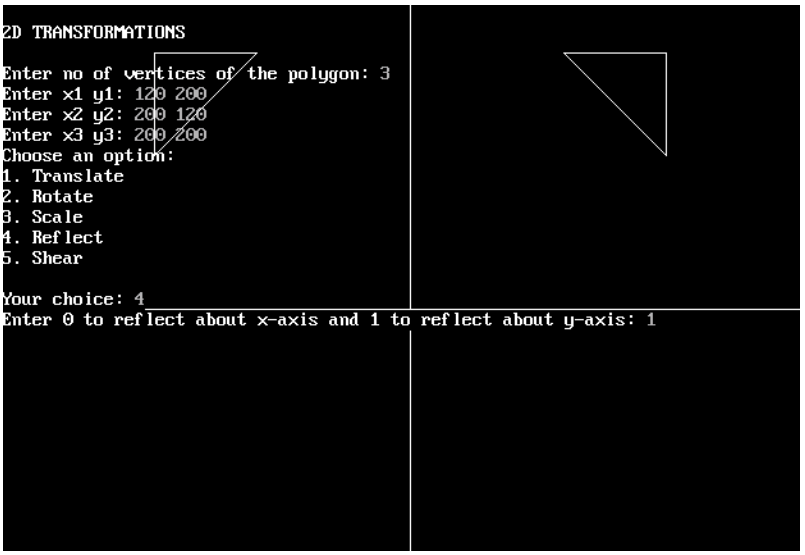
```
    input[2 * i] = xc + x[i];
    input[2 * i + 1] = yc - y[i];
        input[2 * n] = input[0];
        input[2 * n + 1] = input[1];
        drawpoly(n + 1, input);
        getch();
        return 0;
}
```



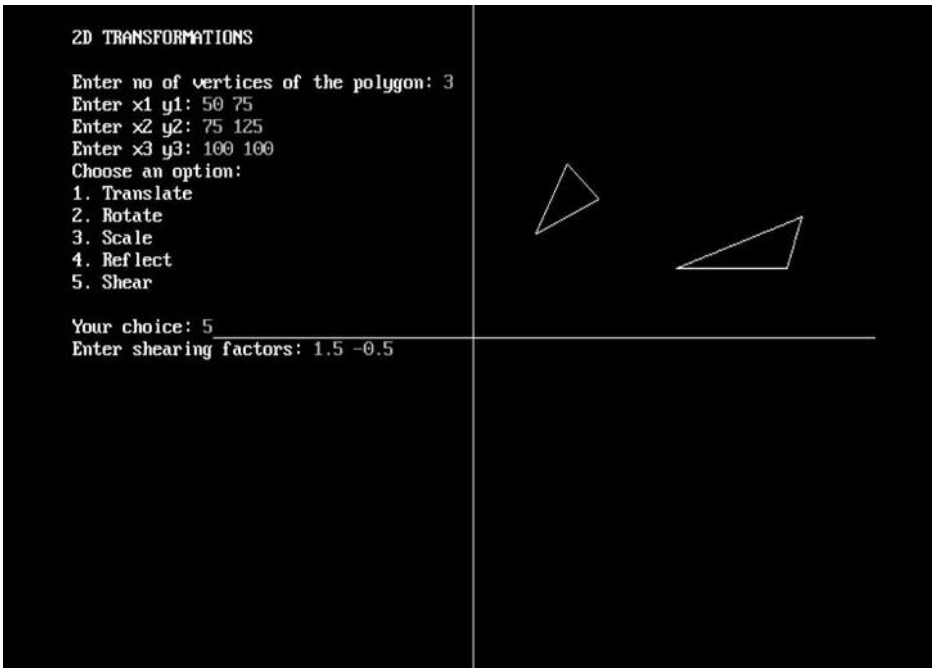
Computer Graphics Lab Experiment No: - 08
Aim: Implement 2D Transformations: Translation, Scaling, Rotation, Reflection, Shear.







Computer Graphics Lab Experiment No: - 08
Aim: Implement 2D Transformations: Translation, Scaling, Rotation, Reflection, Shear.



Conclusion: