UNIT - 2 TRANSFORMATION IN 2D

Scaling:

It is used to alter 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 S_y in y-direction. If the original position is x and y. Scaling factors are S_x and S_y then the value of coordinates after scaling will be x^1 and y_1.

If the picture to be enlarged to twice its original size then $S_x = S_y = 2$. If $S_x$and S_y are not equal then scaling will occur but it will elongate or distort the picture.

If scaling factors are less than one, then the size of the object will be reduced. If scaling factors are higher than one, then the size of the object will be enlarged.

If $S_x$and $S_y$are equal it is also called as Uniform Scaling. If not equal then called as Differential Scaling. If scaling factors with values less than one will move the object closer to coordinate origin, while a value higher than one will move coordinate position farther from origin.

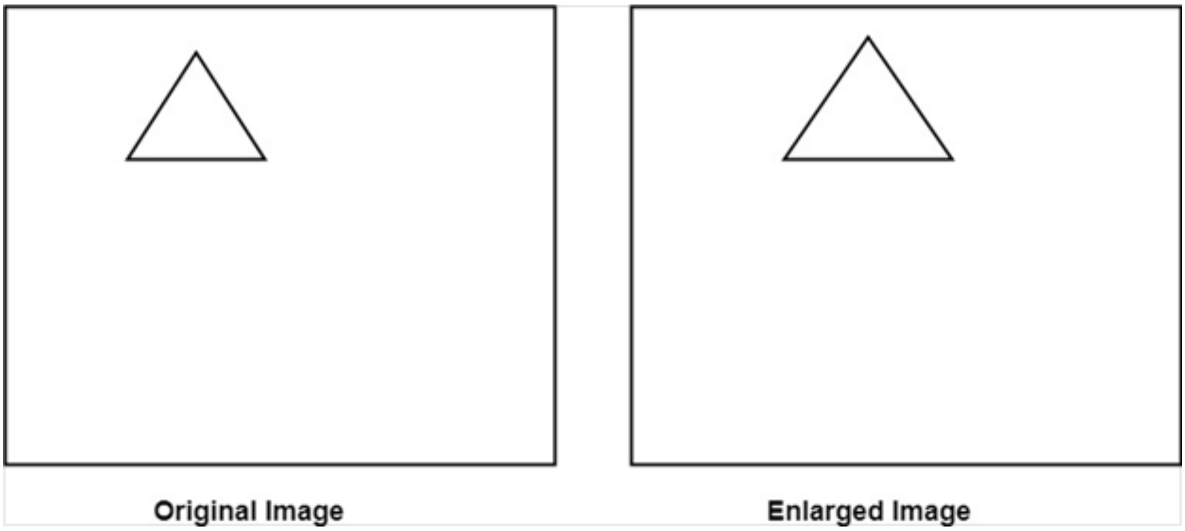
Enlargement: If T_1=

$$\begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$$

,If (x_1 y_1)is original position and $T_1$is translation vector then (x_2 y_2) are coordinated after scaling

$$[x_2 y_2] = [\ x_1 y_1] \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix} = [2x_1 2y_1]$$

The image will be enlarged two times



Reduction: If T_1=

$$\begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix}$$

. If (x_1 y_1) is original position and T_1 is translation vector, then (x_2 y_2) are coordinates after scaling

$$[\,x_2 y_2] = [\ x_1 y_1] \begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix} = [.5x_1 . 5y_1]$$

Translation

It is the straight line movement of an object from one position to another is called Translation. Here the object is positioned from one coordinate location to another.

Translation of point:

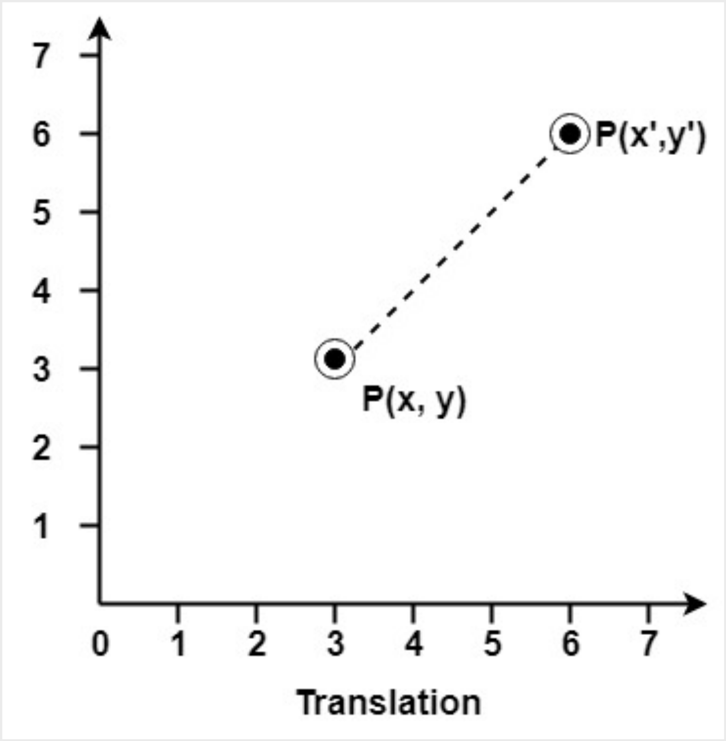
To translate a point from coordinate position (x, y) to another (x_1 y_1), we add algebraically the translation distances T_x and T_y to original coordinate.

The translation pair (T_x,T_y) is called as shift vector.

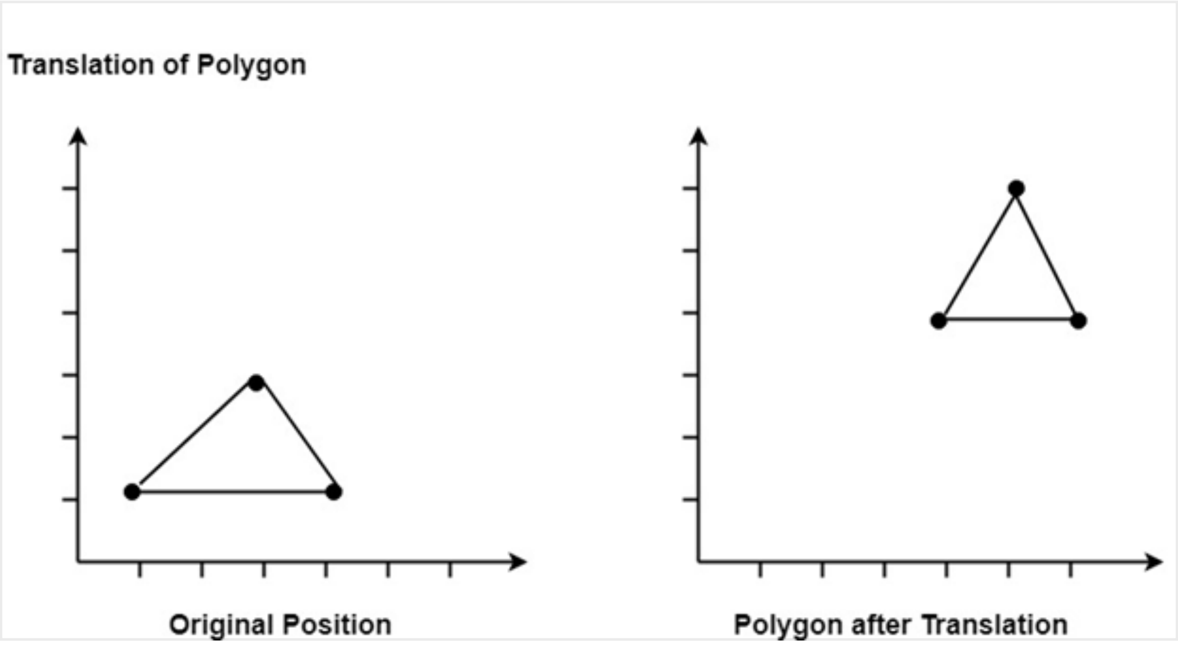
Translation is a movement of objects without deformation. Every position or point is translated by the same amount. When the straight line is translated, then it will be drawn using endpoints.

For translating polygon, each vertex of the polygon is converted to a new position. Similarly, curved objects are translated. To change the position of the circle or ellipse its center coordinates are transformed, then the object is drawn using new coordinates.

Let P is a point with coordinates (x, y). It will be translated as (x^1 y^1).



Translation

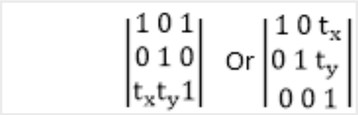


Translation of Polygon

Original Position

Polygon after Translation

Matrix for Translation:



$$\begin{vmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ t_x & t_y & 1 \end{vmatrix} \text{ Or } \begin{vmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{vmatrix}$$

Rotation:

It is a 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 rotation point. Rotation point is also called a pivot point. It is print about which object is rotated.

1. Anticlockwise
2. Counterclockwise

The positive value of the pivot point (rotation angle) rotates an object in a counter-clockwise (anti-clockwise) direction.

The negative value of the pivot point (rotation angle) rotates an object in a clockwise direction.

When the object is rotated, then every point of the object is rotated by the same angle.

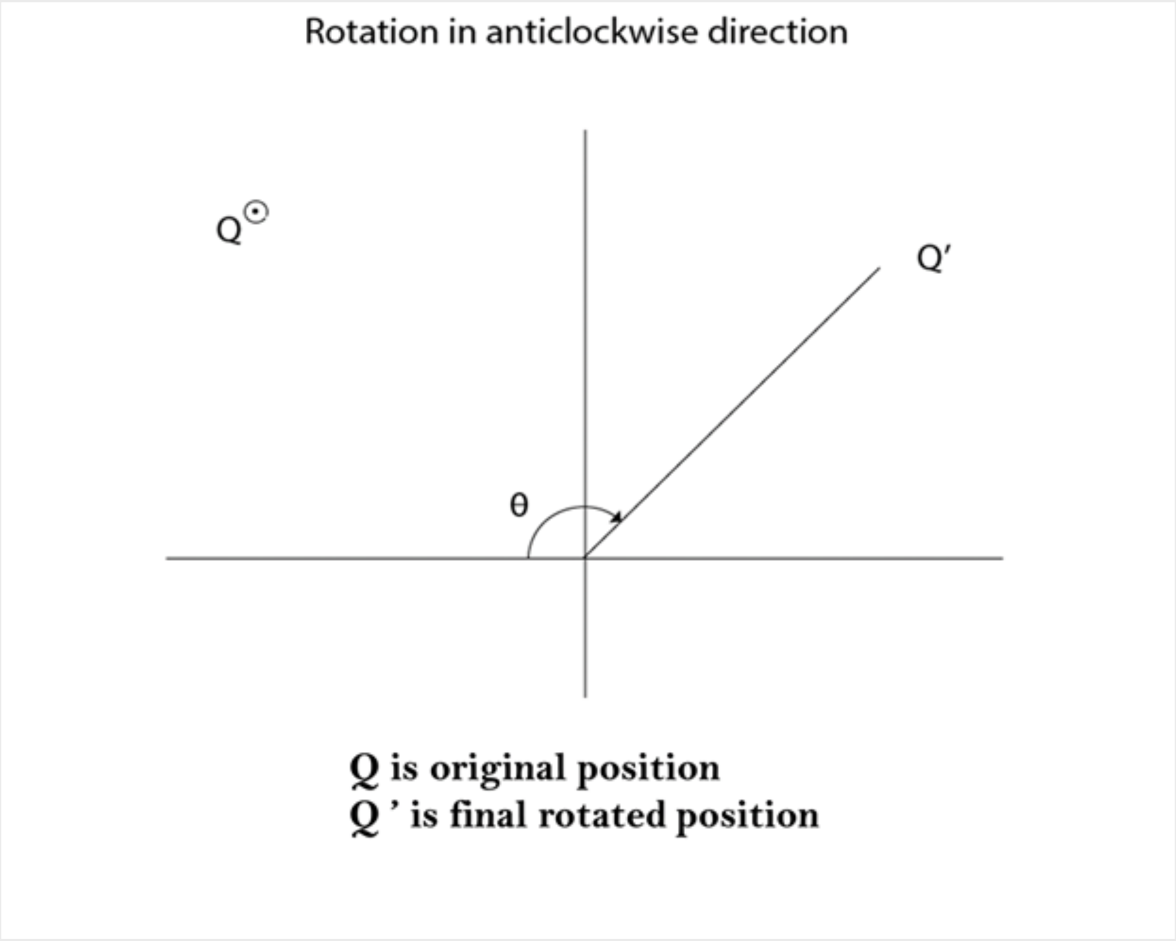
Straight Line: Straight Line is rotated by the endpoints with the same angle and redrawing the line between new endpoints.

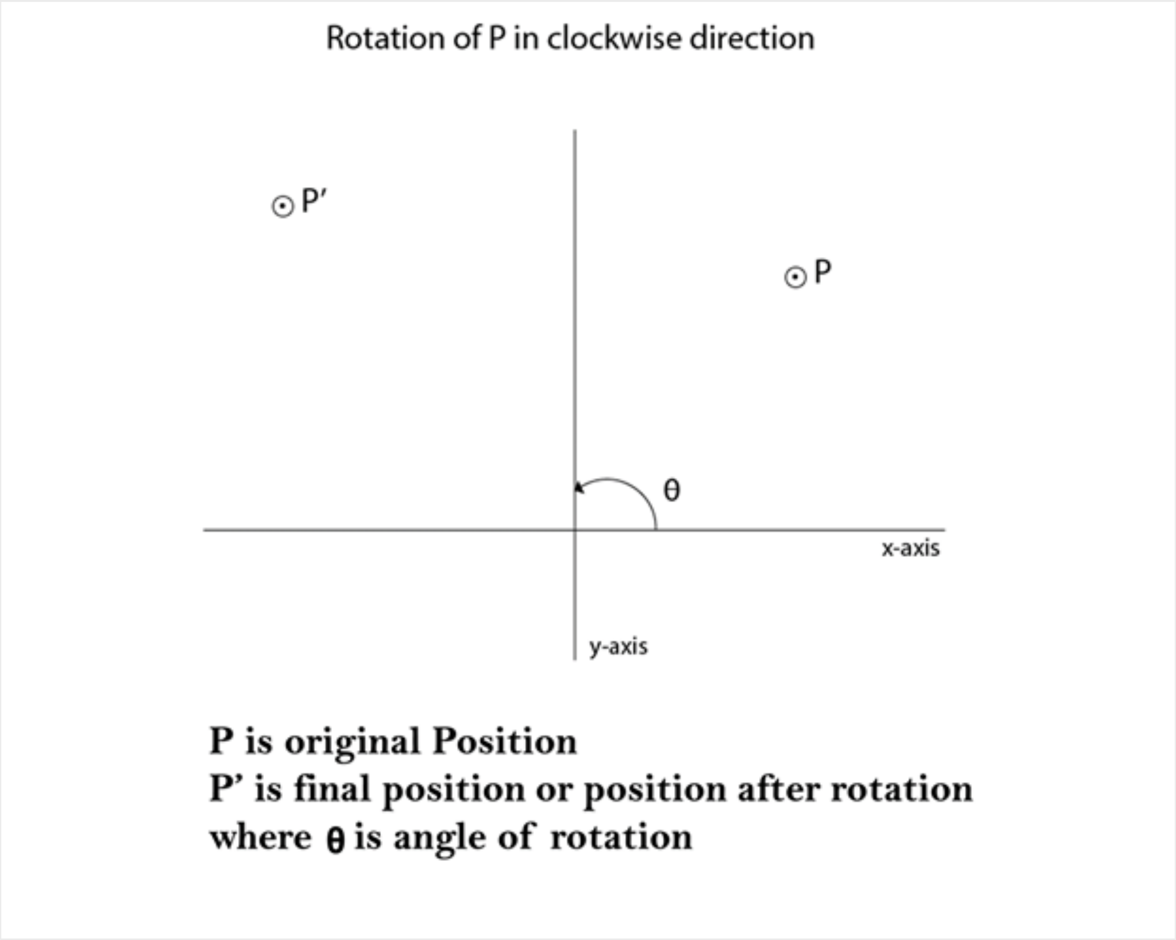
Polygon: Polygon is rotated by shifting every vertex using the same rotational angle.

Curved Lines: Curved Lines are rotated by repositioning of all points and drawing of the curve at new positions.

Circle: It can be obtained by center position by the specified angle.

Ellipse: Its rotation can be obtained by rotating major and minor axis of an ellipse by the desired angle.


Rotation in anticlockwise direction
Q
Q'
θ
Q is original position
Q ' is final rotated position



P is original Position
P’ is final position or position after rotation
where θ is angle of rotation

Matrix for rotation is a clockwise direction.

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

Matrix for rotation is an anticlockwise direction.

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

Matrix for homogeneous co-ordinate rotation (clockwise) angle will become -theta

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

Matrix for homogeneous co-ordinate rotation (anticlockwise)

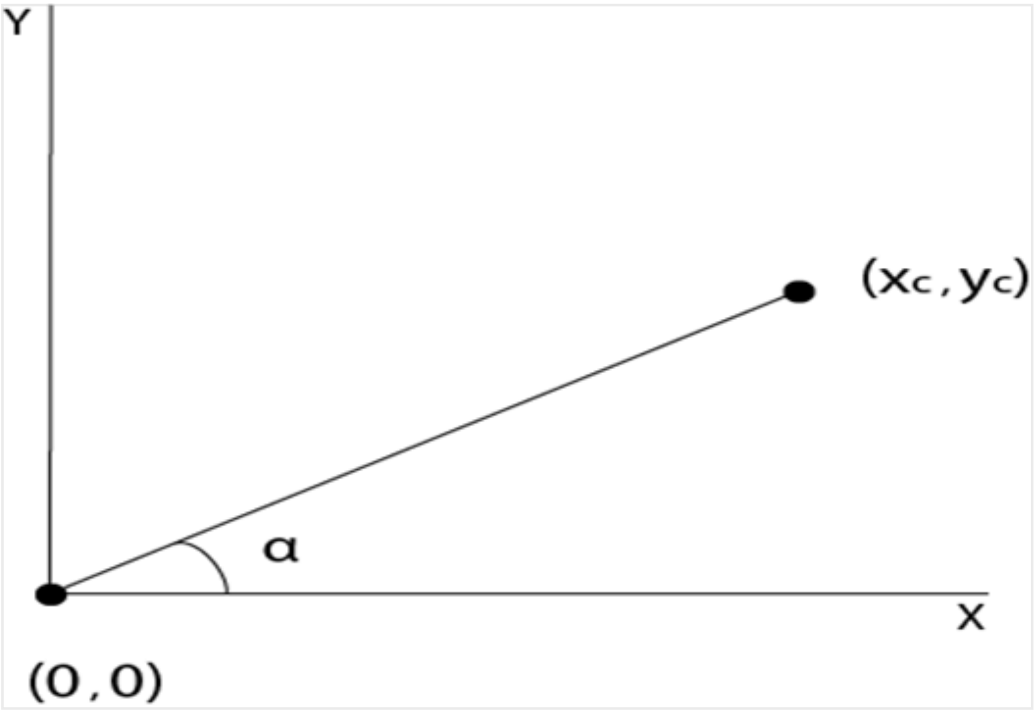
$$R=\begin{pmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

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.

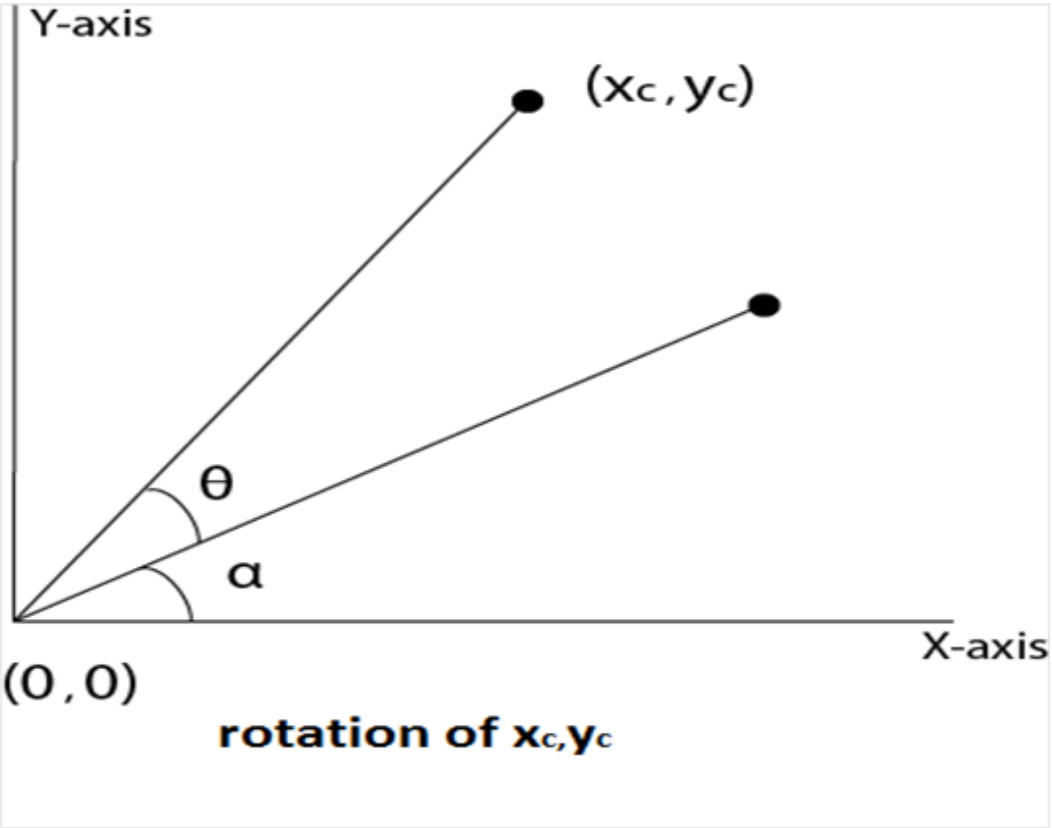
Example: The point (x, y) is to be rotated

The $(x_c\ y_c)$ is a point about which counterclockwise rotation is done

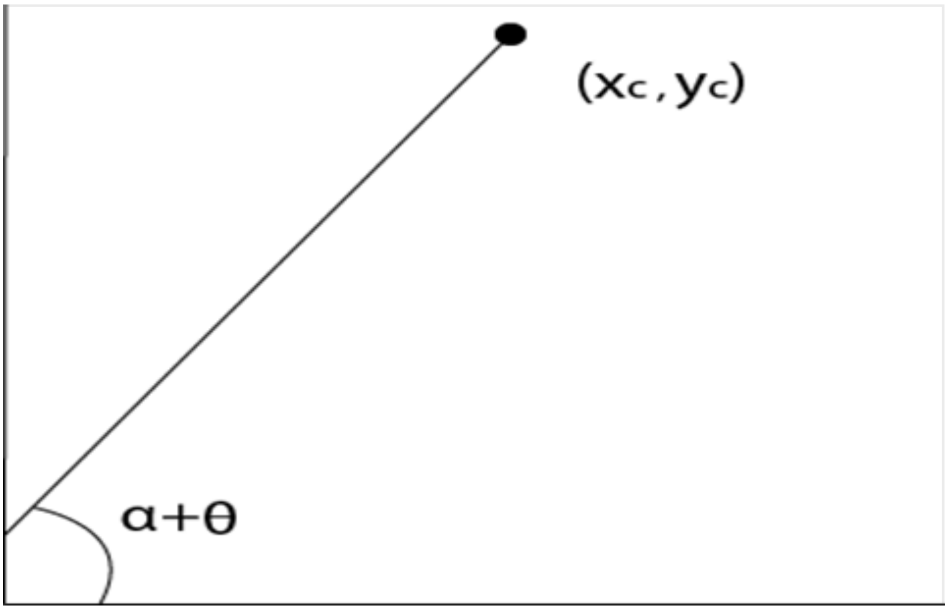
Step1: Translate point $(x_c\ y_c)$ to origin

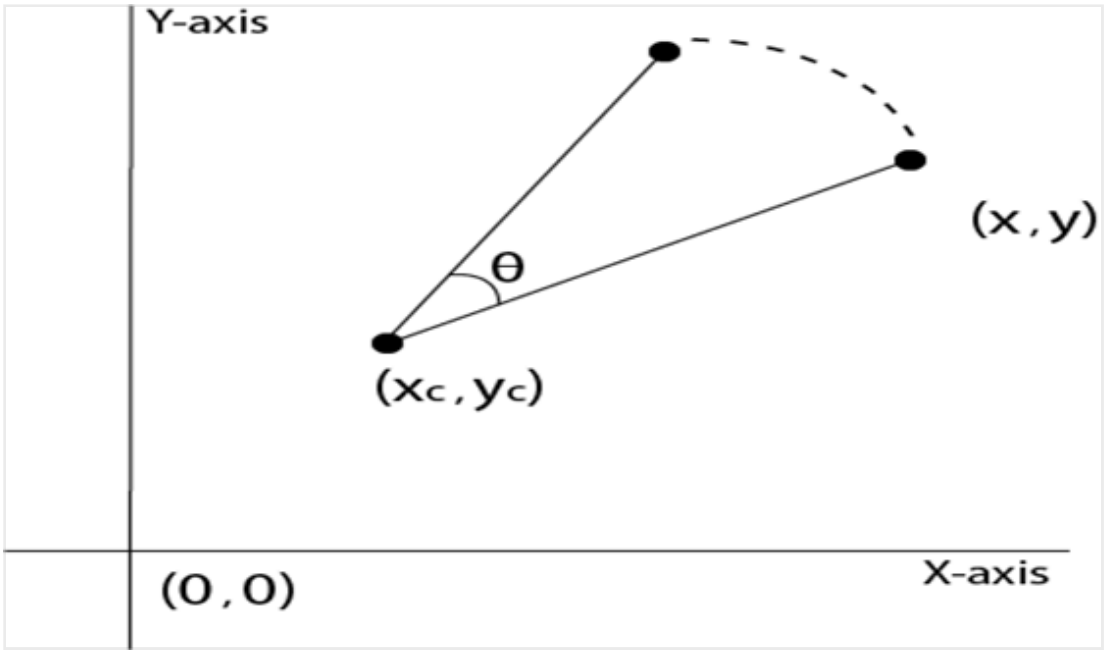


Step2: Rotation of (x, y) about the origin



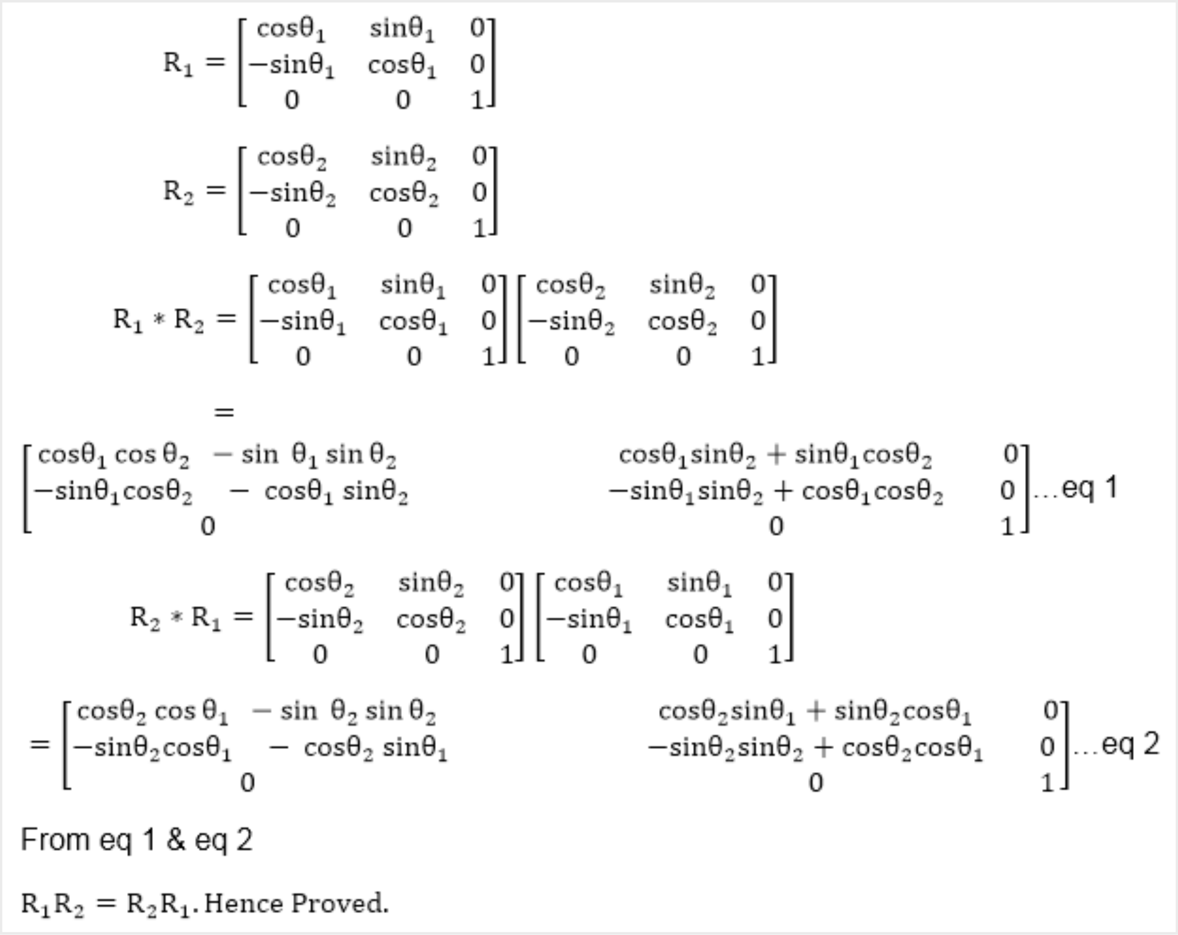
Step3: Translation of center of rotation back to its original position





Example1: Prove that 2D rotations about the origin are commutative i.e. R_1 R_2=R_2 R_1.

Solution: R_1 and $R_2$are rotation matrices



$$R_1 = \begin{bmatrix} \cos\theta_1 & \sin\theta_1 & 0 \\ -\sin\theta_1 & \cos\theta_1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$R_2 = \begin{bmatrix} \cos\theta_2 & \sin\theta_2 & 0 \\ -\sin\theta_2 & \cos\theta_2 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$R_1 * R_2 = \begin{bmatrix} \cos\theta_1 & \sin\theta_1 & 0 \\ -\sin\theta_1 & \cos\theta_1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta_2 & \sin\theta_2 & 0 \\ -\sin\theta_2 & \cos\theta_2 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} \cos\theta_1 \cos\theta_2 - \sin\theta_1 \sin\theta_2 & \cos\theta_1\sin\theta_2 + \sin\theta_1\cos\theta_2 & 0 \\ -\sin\theta_1\cos\theta_2 - \cos\theta_1 \sin\theta_2 & -\sin\theta_1\sin\theta_2 + \cos\theta_1\cos\theta_2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \text{...eq 1}$$

$$R_2 * R_1 = \begin{bmatrix} \cos\theta_2 & \sin\theta_2 & 0 \\ -\sin\theta_2 & \cos\theta_2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta_1 & \sin\theta_1 & 0 \\ -\sin\theta_1 & \cos\theta_1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

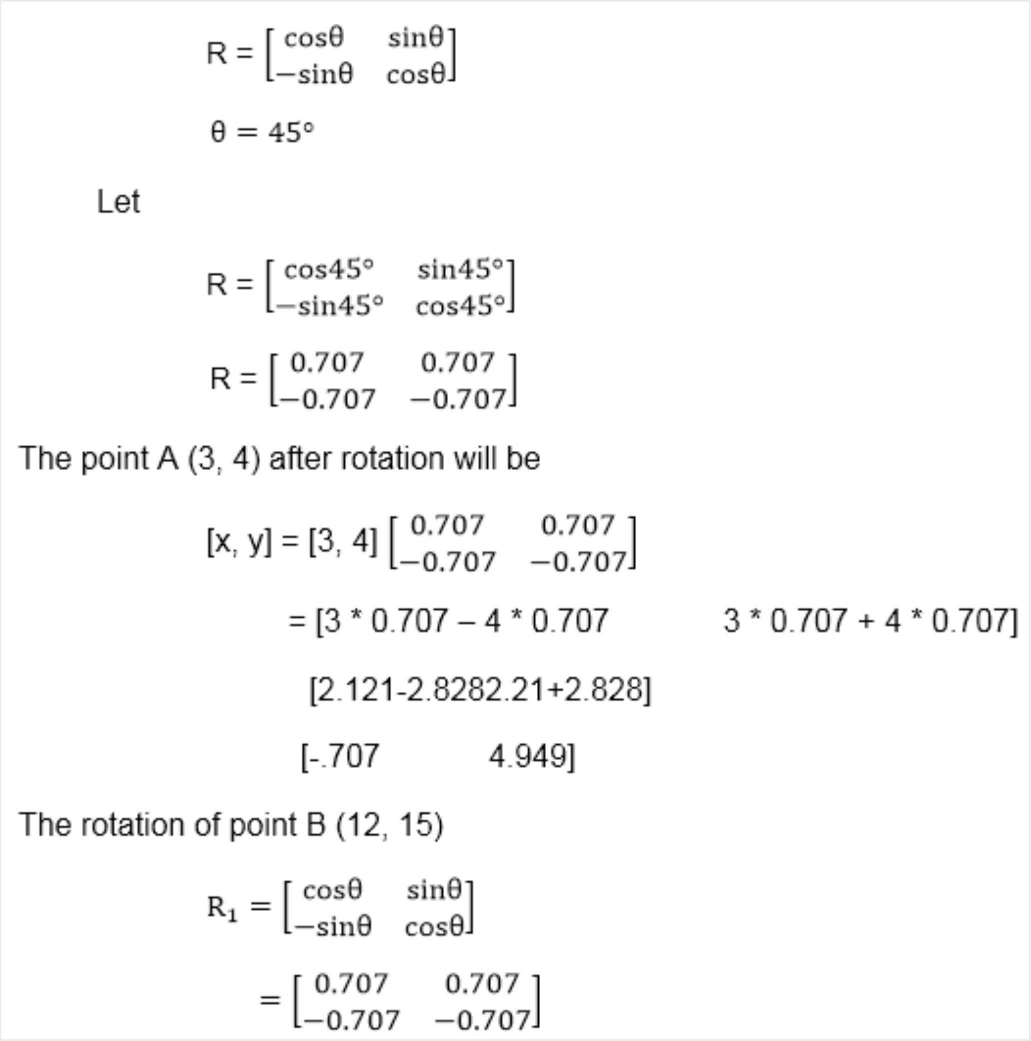
$$= \begin{bmatrix} \cos\theta_2 \cos\theta_1 - \sin\theta_2 \sin\theta_2 & \cos\theta_2\sin\theta_1 + \sin\theta_2\cos\theta_1 & 0 \\ -\sin\theta_2\cos\theta_1 - \cos\theta_2 \sin\theta_1 & -\sin\theta_2\sin\theta_2 + \cos\theta_2\cos\theta_1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \text{...eq 2}$$

From eq 1 & eq 2

$R_1R_2 = R_2R_1$. Hence Proved.

Example2: Rotate a line CD whose endpoints are (3, 4) and (12, 15) about origin through a 45° anticlockwise direction.

Solution: The point C (3, 4)



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

$\theta = 45°$

Let

R = $\begin{bmatrix} \cos 45° & \sin 45° \\ -\sin 45° & \cos 45° \end{bmatrix}$

R = $\begin{bmatrix} 0.707 & 0.707 \\ -0.707 & -0.707 \end{bmatrix}$

The point A (3, 4) after rotation will be

[x, y] = [3, 4] $\begin{bmatrix} 0.707 & 0.707 \\ -0.707 & -0.707 \end{bmatrix}$

= [3 * 0.707 – 4 * 0.707   3 * 0.707 + 4 * 0.707]

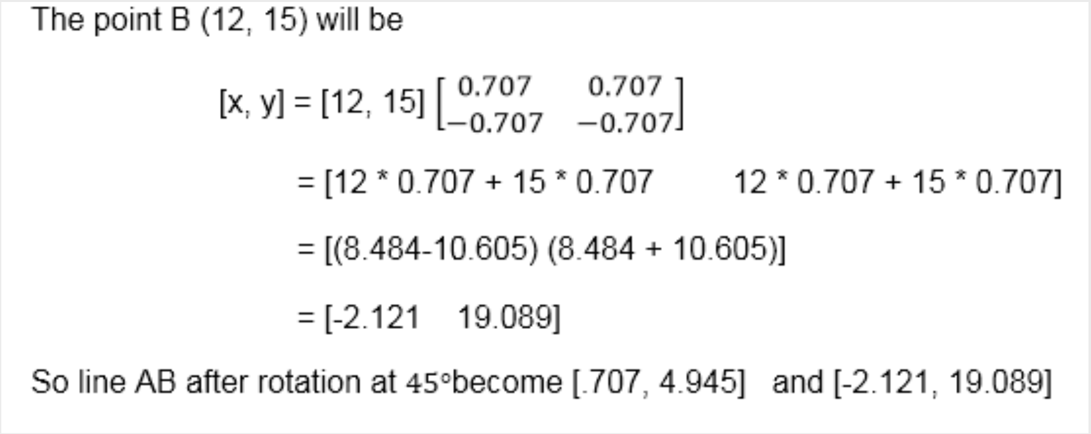
[2.121-2.8282.21+2.828]

[-.707   4.949]

The rotation of point B (12, 15)

$R_1 = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix}$

$= \begin{bmatrix} 0.707 & 0.707 \\ -0.707 & -0.707 \end{bmatrix}$



The point B (12, 15) will be

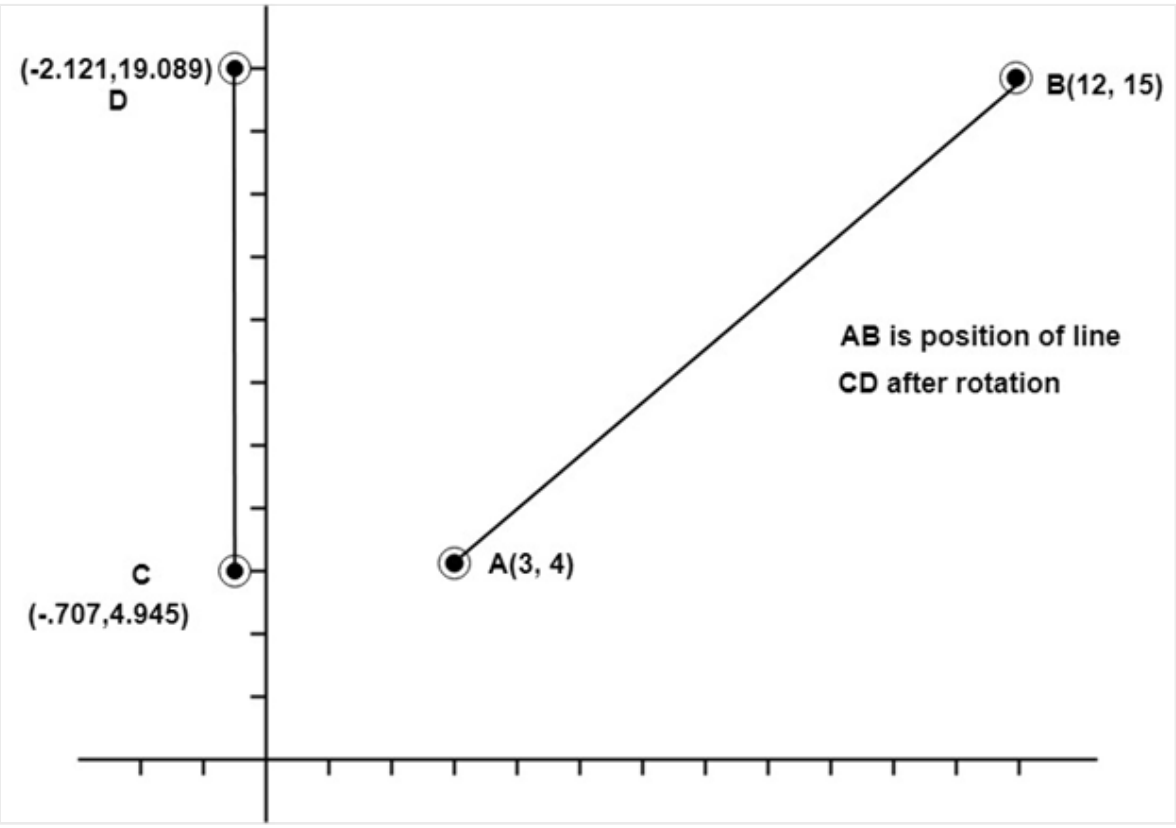
[x, y] = [12, 15] $\begin{bmatrix} 0.707 & 0.707 \\ -0.707 & -0.707 \end{bmatrix}$

= [12 * 0.707 + 15 * 0.707   12 * 0.707 + 15 * 0.707]

= [(8.484-10.605) (8.484 + 10.605)]

= [-2.121   19.089]

So line AB after rotation at 45°become [.707, 4.945]   and [-2.121, 19.089]



Example3: Rotate line AB whose endpoints are A (2, 5) and B (6, 12) about origin through a 30° clockwise direction.

Solution: For rotation in the clockwise direction. The matrix is

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

Step1: Rotation of point A (2, 5). Take angle 30°

$$R = \begin{bmatrix} \cos 30^\circ & -\sin 30^\circ \\ \sin 30^\circ & \cos 30^\circ \end{bmatrix}$$

$$= [2, 5] \begin{bmatrix} \cos 30^\circ & -\sin 30^\circ \\ -\sin 30^\circ & \cos 30^\circ \end{bmatrix}$$

$$= [2, 5] \begin{bmatrix} .866 & -0.5 \\ .5 & .866 \end{bmatrix}$$

$$= [2 * .866 + 5 * .5 \qquad 2 * (-.5) + 5\ (.866)]$$

$$= [(1.732 + 2.5 \qquad -1 + 4.33]$$

$$= [4.232 \qquad 3.33]$$

A point (2, 5) become (4.232, 3.33)

Step2: Rotation of point B (6, 12)

$$= [6 \quad 12] \begin{bmatrix} \cos 30^\circ & -\sin 30^\circ \\ \sin 30^\circ & \cos 30^\circ \end{bmatrix}$$

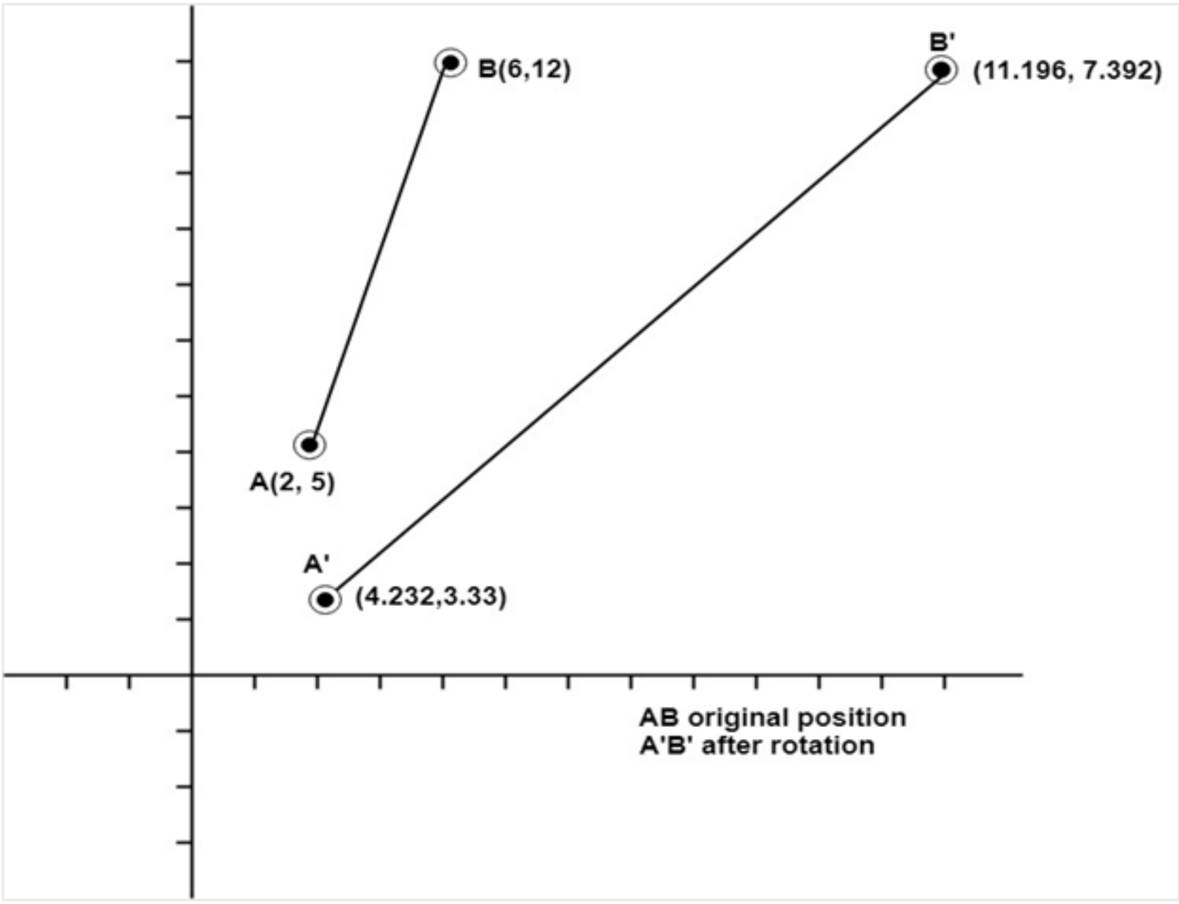
$$= [6 \; 12] \begin{bmatrix} .866 & -0.5 \\ .5 & .866 \end{bmatrix}$$

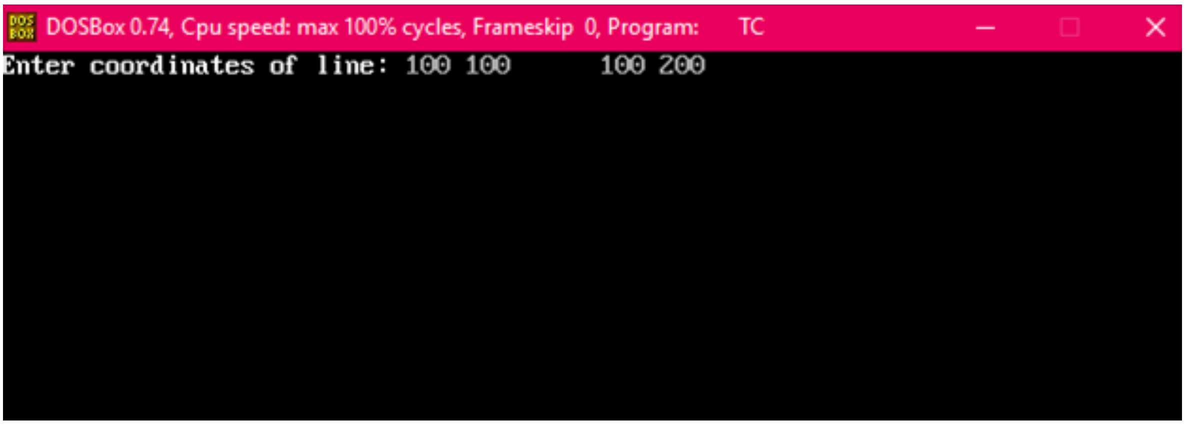
$$= [6 * .866 + 12 * .5 \qquad 6 * (-.5) + 12 * (.866)]$$

$$= [5.196 + 6 \qquad -3 + 10.392]$$

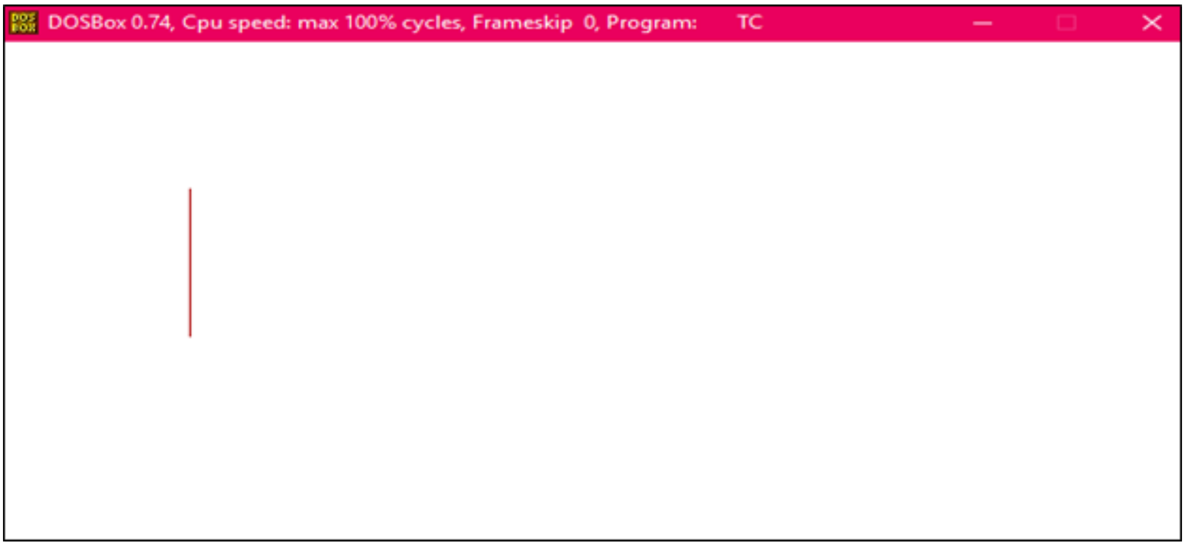
$$= [11.196 \qquad 7.392]$$

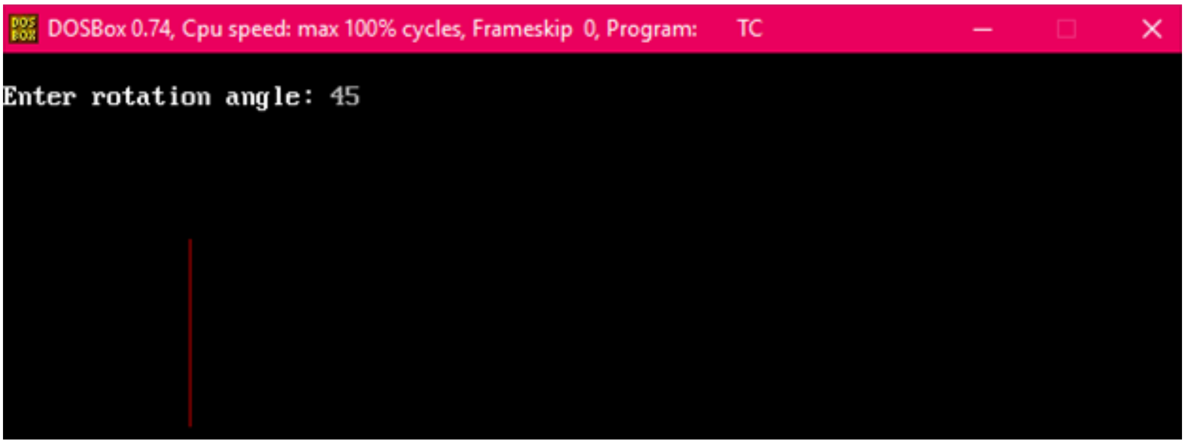
B point (6, 12) becomes (11.46, 7.312) after rotation 30° in clockwise direction.




DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Program: TC

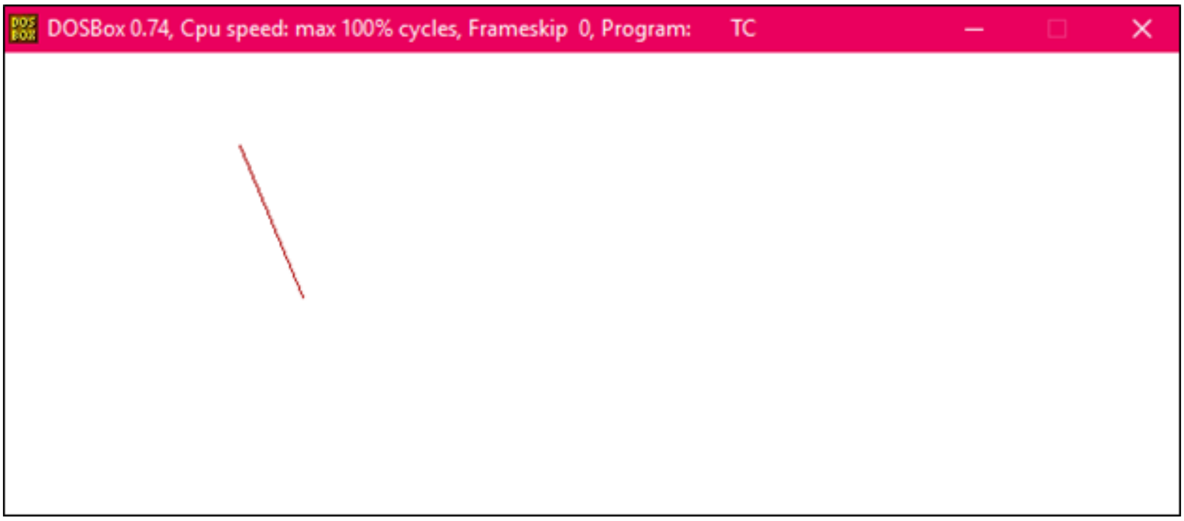
```
Enter coordinates of line: 100 100      100 200
```


DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Program: TC

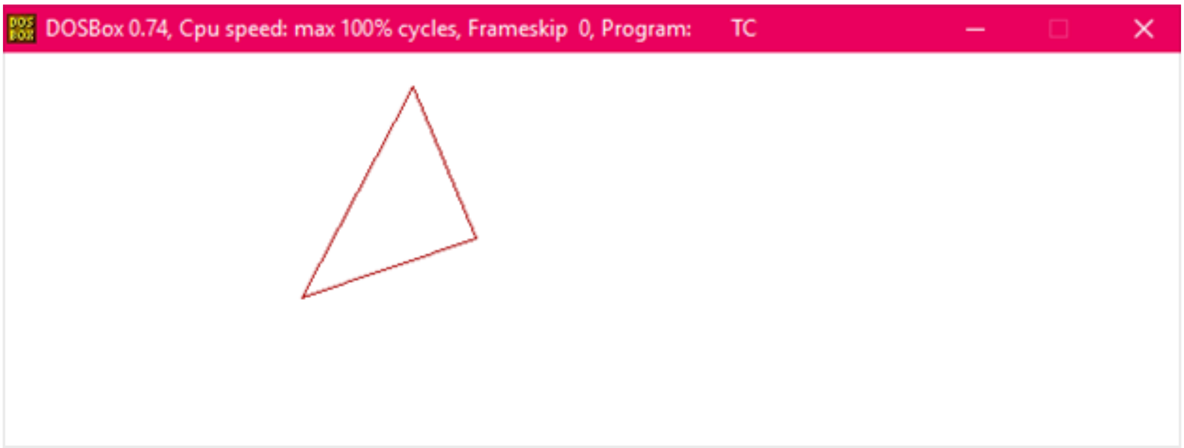

DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Program: TC

```
Enter rotation angle: 45
```

After rotation


DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Program: TC

After rotation


DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Program: TC

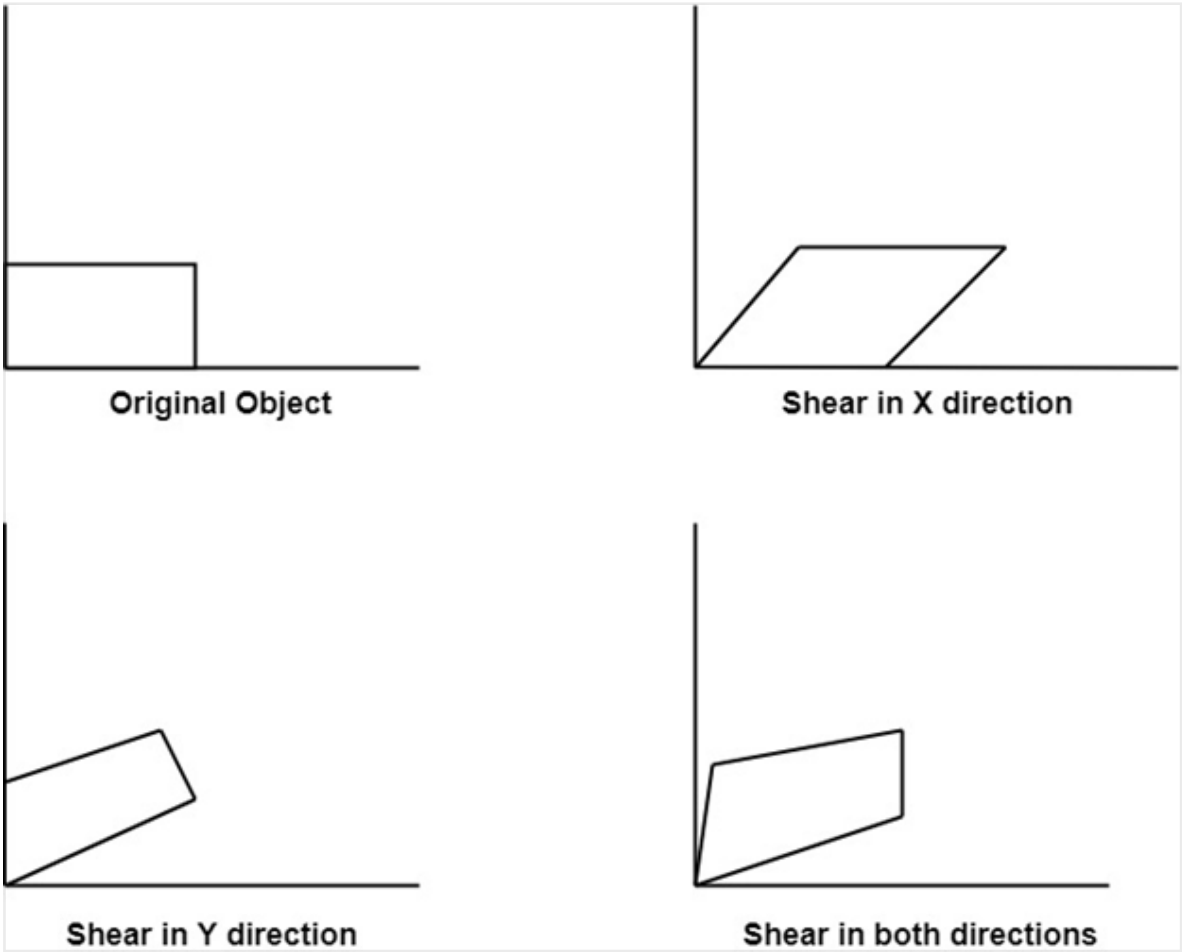
REFLECTION & SHEARING

Shearing:

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

Shearing in the X-direction: In this horizontal shearing sliding of layers occur. The homogeneous matrix for shearing in the x-direction is shown below:

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



Original Object

Shear in X direction

Shear in Y direction

Shear in both directions

Shearing in the Y-direction: Here shearing is done by sliding along vertical or y-axis.

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

Shearing in X-Y directions: 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:

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

Reflection:

It is a transformation which produces a mirror image of an object. The mirror image can be either about x-axis or y-axis. The object is rotated by180°.

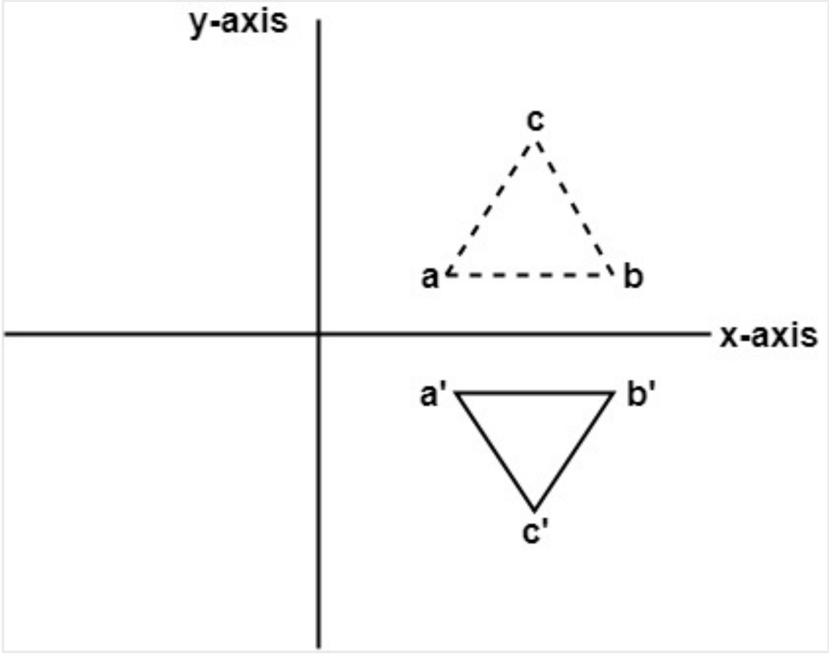
Types of Reflection:

1. Reflection about the x-axis
2. Reflection about the y-axis
3. Reflection about an axis perpendicular to xy plane and passing through the origin
4. Reflection about line y=x

1. Reflection about x-axis: The object can be reflected about x-axis with the help of the following matrix

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

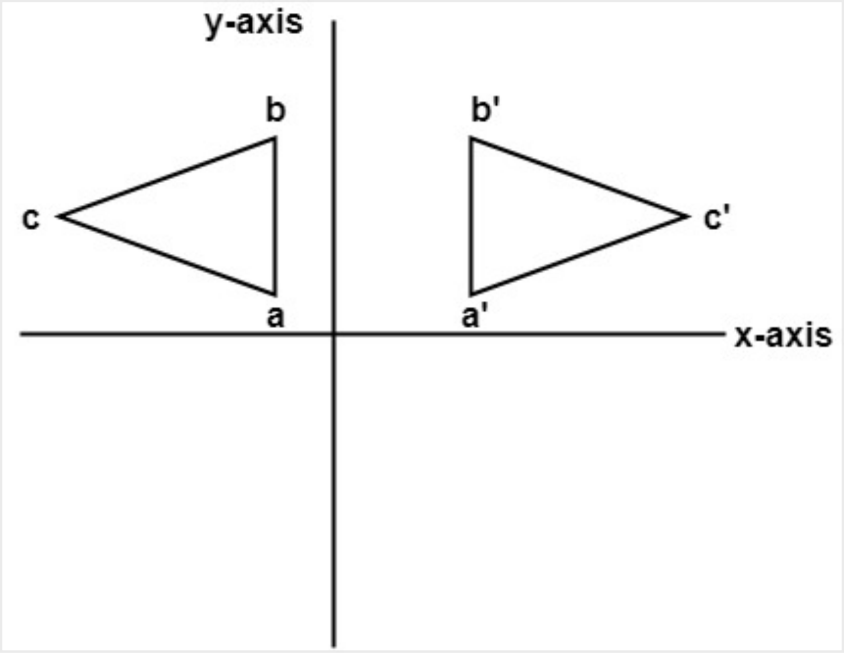
In this transformation value of x will remain same whereas the value of y will become negative. Following figures shows the reflection of the object axis. The object will lie another side of the x-axis.



2. Reflection about y-axis: The object can be reflected about y-axis with the help of following transformation matrix

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

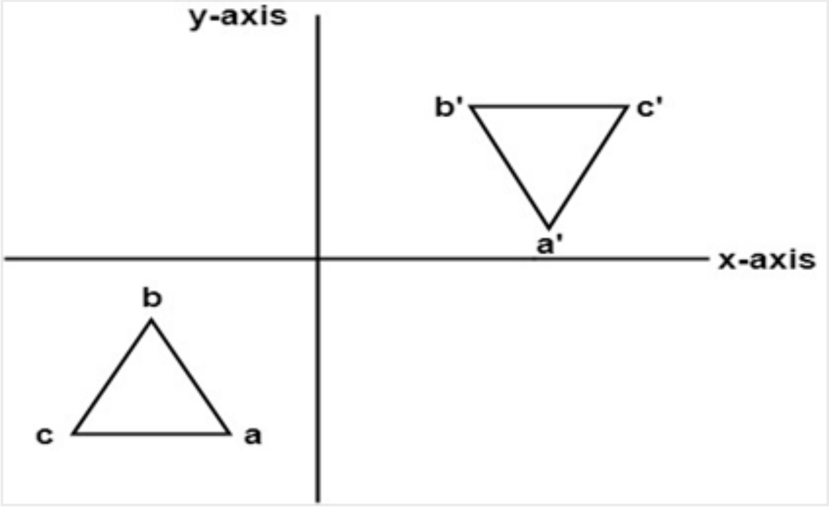
Here the values of x will be reversed, whereas the value of y will remain the same. The object will lie another side of the y-axis. The following figure shows the reflection about the y-axis



3. Reflection about an axis perpendicular to xy plane and passing through origin:

In the matrix of this transformation is given below

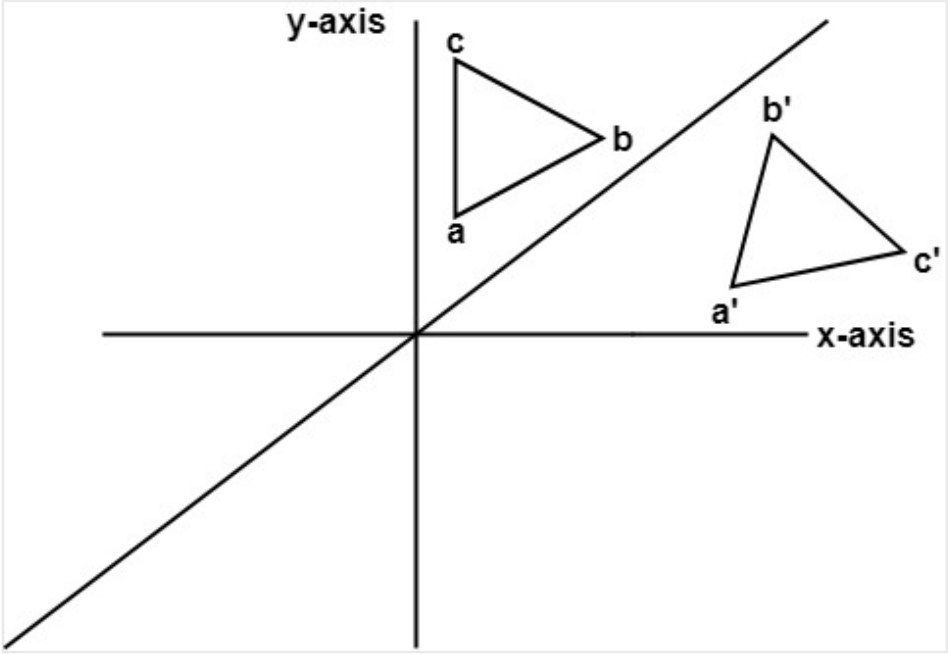
$$\begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$



In this value of x and y both will be reversed. This is also called as half revolution about the origin.

4. Reflection about line y=x: The object may be reflected about line y = x with the help of following transformation matrix

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



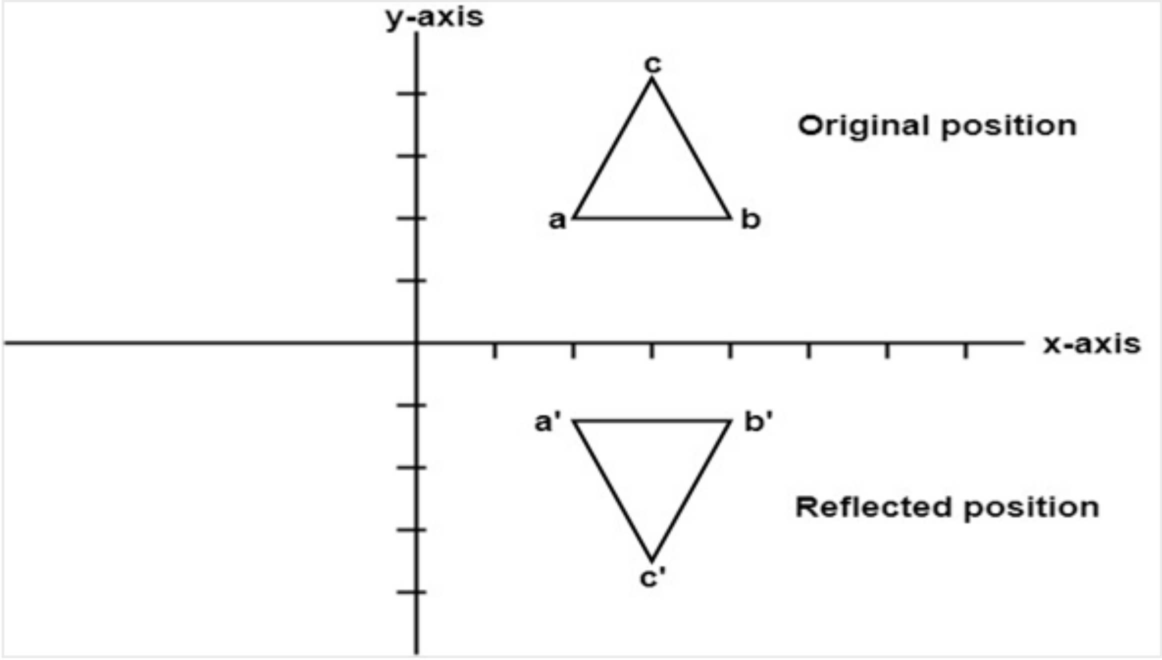
First of all, the object is rotated at 45°. The direction of rotation is clockwise. After it reflection is done concerning x-axis. The last step is the rotation of y=x back to its original position that is counterclockwise at 45°.

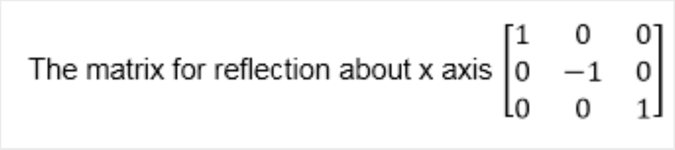
Example: A triangle ABC is given. The coordinates of A, B, C are given as

A	(3 4)
B	(6 4)
C	(4 8)

Find reflected position of triangle i.e., to the x-axis.

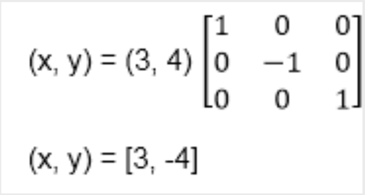
Solution:





The matrix for reflection about x axis $\begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

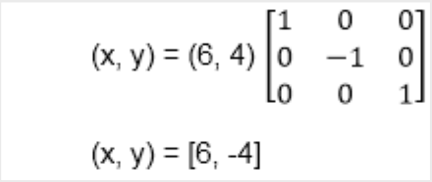
The a point coordinates after reflection



$(x, y) = (3, 4) \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

(x, y) = [3, -4]

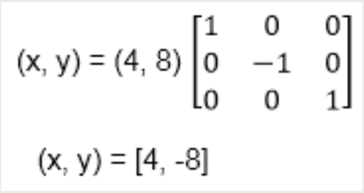
The b point coordinates after reflection



$(x, y) = (6, 4) \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

(x, y) = [6, -4]

The coordinate of point c after reflection



$(x, y) = (4, 8) \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

(x, y) = [4, -8]

a (3, 4) becomes a^1 (3, -4)

b (6, 4) becomes b^1 (6, -4)

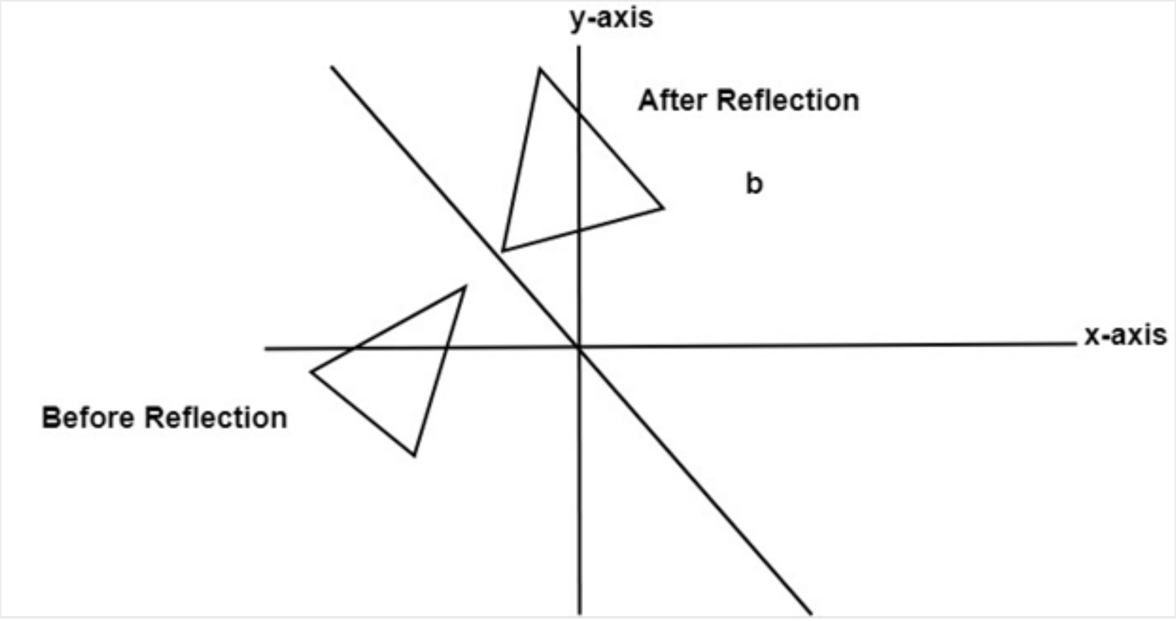
c (4, 8) becomes c^1 (4, -8)

Program to perform Mirror Reflection about a line:

```
#include <iostream.h>
#include <conio.h>
#include <graphics.h>
#include <math.h>
#include <stdlib.h>
#define pi 3.14
class arc
{
    float x[10],y[10],theta,ref[10][10],ang;
            float p[10][10],p1[10][10],x1[10],y1[10],xm,ym;
    int i,k,j,n;
    public:
    void get();
    void cal ();
    void map ();
    void graph ();
    void plot ();
    void plot1();
};
void arc::get ()
{
    cout<<"\n ENTER ANGLE OF LINE INCLINATION AND Y INTERCEPT";
    cin>> ang >> b;
    cout <<"\n ENTER NO OF VERTICES";
    cin >> n;
    cout <<"\n ENTER";
    for (i=0; i<n; i++)
    {
        cout<<"\n x["<<i<<"] and y["<<i<<"]";
    }
    theta =(ang * pi)/ 180;
    ref [0] [0] = cos (2 * theta);
    ref [0] [1] = sin (2 * theta);
    ref [0] [2] = -b *sin (2 * theta);
    ref [1] [0] = sin (2 * theta);
    ref [1] [1] = -cos (2 * theta);
    ref [1] [2] = b * (cos (2 * theta)+1);
    ref [2] [0]=0;
    ref [2] [1]=0;
    ref [2] [2] = 1;
}
void arc :: cal ()
{
    for (i=0; i < n; i++)
    {
        p[0] [i] = x [i];
        p [1] [i] = y [i];
        p [2] [i] = 1;
    }
    for (i=0; i<3;i++)
    {
        for (j=0; j<n; j++)
        {
          p1 [i] [j]=0;
          for (k=0;k<3; k++)
       }
       p1 [i] [j] + = ref [i] [k] * p [k] [j];
           }
for (i=0; i<n; i++)
  {
   x1 [i]=p1[0] [i];
   y1 [i] = p1 [1] [i];
   }
}
void arc :: map ()
{
   int gd = DETECT,gm;
   initgraph (&gd, &gm, " ");
          int errorcode = graphresult ();

   if (errorcode ! = grOK)
   {
       printf ("Graphics error: %s \n", grapherrormsg (errorcode));
       printf ("Press any key to halt:");
       getch ();
       exit (1);
   }
}
void arc :: graph ()
{
   xm=getmaxx ()/2;
   ym=getmaxy ()/2;
   line (xm, 0, xmm 2*ym);
}
void arc :: plot 1 ()
{
   for (i=0; i <n-1; i++)
   {
       circle (x1[i]+xm, (-y1[i]+ym), 2);
       line (x1[i]+xm, (-y1[i]+ym), x1[i+1]+xm, (-y1[i+1]+ym));
   }
       line (x1[n-1)+xm, (-y1[n-1]+ym), x1[0]+xm, (-
y1[0]+ym));
        getch();
}
void arc :: plot ()
{
    for (i=0; i <n-1; i++)
    {
        circle (x1[i]+xm, (-y1[i]+ym, 2);
        line (x1[i]+xm, (-y1[i]+ym), x[i+1]+xm, (-y1[i+1]+ym));
    }
        line (x[n-1]+xm, (-y1[n-1]+ym), x[0]+xm, (-y[0]+ym));
        getch();
}
void main ()
{
    class arc a;
    clrscr();
    a.map();
    a.graph();
    a.get();
    a.cal();
    a.plot();
    a.plot1();
    getch();
}
```

Output:


y-axis
After Reflection
b
x-axis
Before Reflection