Expt No. B4

Assignment No. 8



**Title** :- Reflection & rotation about arbitary points.

**Problem statement** :- Write C++ / Java program to implement reflection of 2-D object about x-axis, y axis & about x=y axis. Also rotate object about arbitary point given by user.

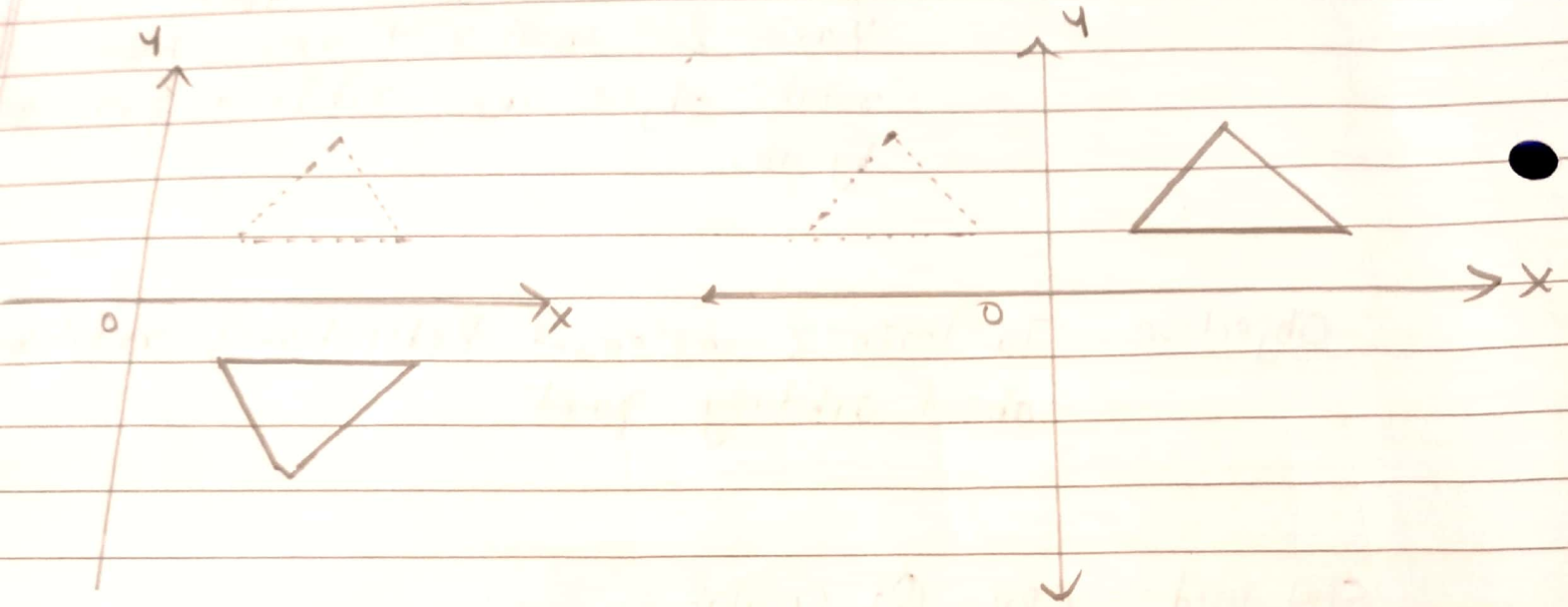
**Objective** :- To learn & implement Reflection & rotation about arbitary point.

**SW used** :- Cpp, Qt creator.

**Theory** :-

- Reflection is mirror image of original object. In other words, we can say that it is a rotation operation with 180°.
- In reflection transformation, the size of object does not change.

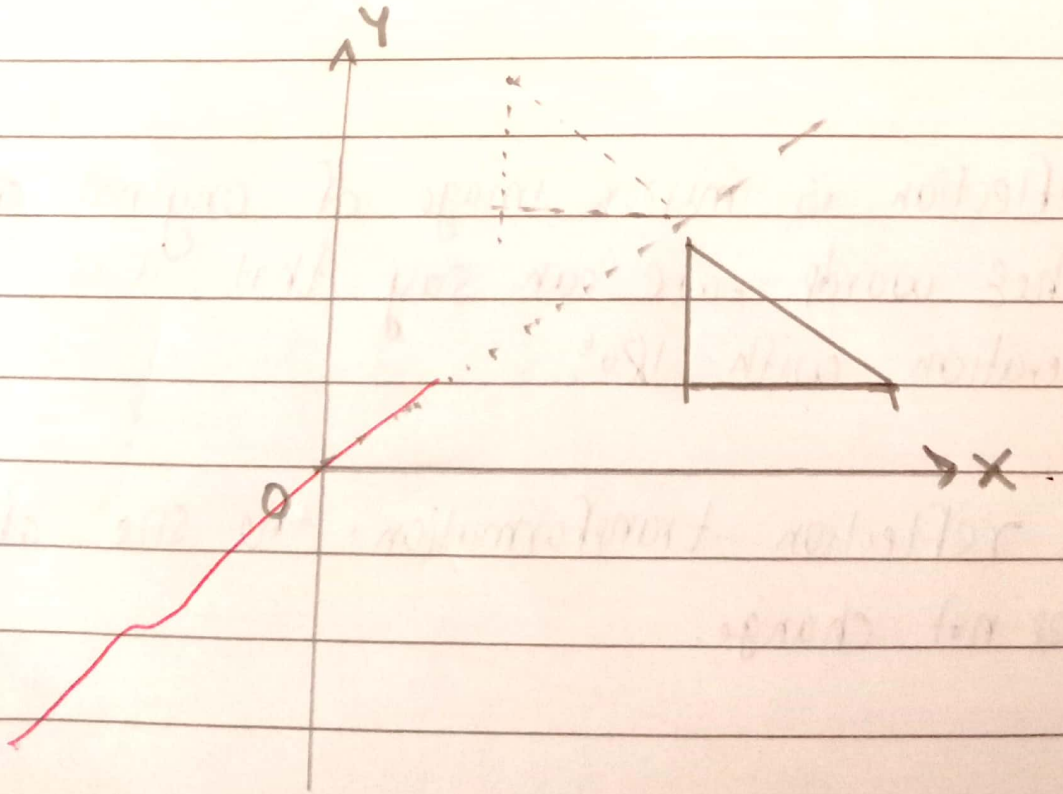
- **Reflection can be done**
  - with respect to x-axis
  - with respect to y-axis
  - with $x = y$




Reflection about x-axis




Reflection about y-axis



Reflection about x-axis

$$P = P' * T$$

$$\begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} x_1' \\ y_1' \\ 1 \end{bmatrix}$$

Reflection about y-axis :-

$$P = P' * T$$

$$\begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} x_1' \\ y_1' \\ 1 \end{bmatrix}$$

Reflection w.r.t $x = y$

$$\begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$P = P' * T$$

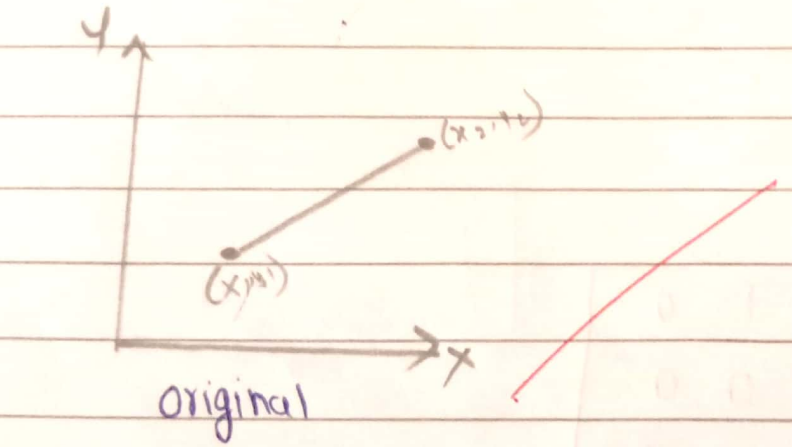
- **Rotation about arbitary point:**

Following steps are needed to perform the rotation about arbitary points

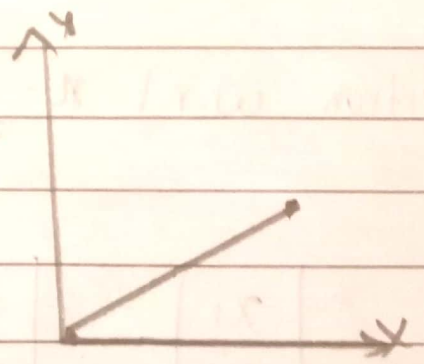
- Translate the center origin point
- Perform required rotation.
- Translate back to arbitary points.

$$[x^* \; y^* \; 1] = [x \; y \; 1] \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -m & -n & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ m & n & 1 \end{bmatrix}$$

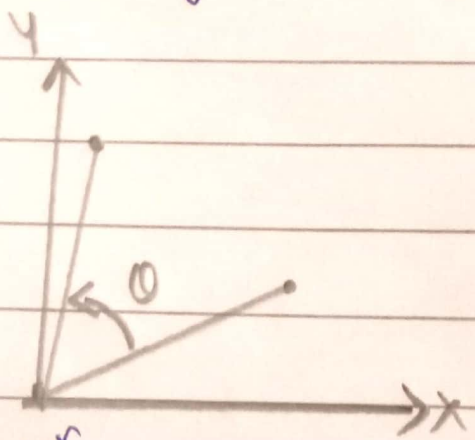
transformed matrix | coordinate matrix | translation matrix | Rotation matrix | translation matrix



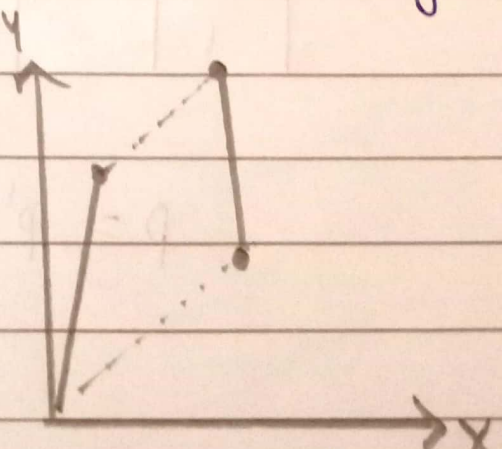

original




Translate to origin




perform required rotation




translate back to origi

Pseudo code for multiplication matrices

```
procedure MatrixMulti ( CM, TM)
    input  CM , TM
    output  TxM

  for (i=0; i<n; i++)
     for (j=0;  j<n; j++)
          TxM[i][j] = 0
          for (k=0; k<n; k++)

          TxM[i][j]= TxM[i][j] + (CM[i][k] * TxM[k][j]);

          end for
     end for
  end for
end procedure.
```

Conclusion :-

We have learn & implement reflection about x, y
x=y axis & Rotation about arbitrary points.