**­LAB ASSIGNMENT 6**

**Objective : Implement 2D reflection and shearing transformations on geometric shapes. Tasks:**

* **Flip the point or shape across the x-axis.**
* **Flip the point or shape across the y-axis.**
* **Reflect the point or shape across a given line.**
* **Skew the point or shape along the x-axis.**
* **Skew the point or shape along the y-axis.**

**CODE:**

import numpy as np

import matplotlib.pyplot as plt

# Function to plot the shape

def plot\_shape(points, label, color):

points = np.array(points)

x, y = points[:, 0], points[:, 1]

plt.plot(x, y, marker='o', label=label, color=color)

plt.fill(x, y, alpha=0.3, color=color)

# Transformation function using matrix multiplication

def apply\_transformation(points, matrix):

return [matrix.dot([x, y]) for x, y in points]

# Define a rectangle (5 points to close the shape)

rectangle = [[1, 1], [5, 1], [5, 3], [1, 3], [1, 1]]

# Define transformation matrices

flip\_x = np.array([[1, 0],

[0, -1]])

flip\_y = np.array([[-1, 0],

[0, 1]])

reflect\_yx = np.array([[0, 1],

[1, 0]])

shear\_x = np.array([[1, 1.5],

[0, 1]])

shear\_y = np.array([[1, 0],

[0.5, 1]])

# Plot original shape and transformations

plt.figure(figsize=(8, 8))

plot\_shape(rectangle, "Original Rectangle", 'blue')

plot\_shape(apply\_transformation(rectangle, flip\_x), "Flipped X", 'red')

plot\_shape(apply\_transformation(rectangle, flip\_y), "Flipped Y", 'green')

plot\_shape(apply\_transformation(rectangle, reflect\_yx), "Reflected y=x", 'orange')

plot\_shape(apply\_transformation(rectangle, shear\_x), "Sheared X", 'purple')

plot\_shape(apply\_transformation(rectangle, shear\_y), "Sheared Y", 'cyan')

# Show plot with settings

plt.axhline(0, color='black',linewidth=0.5)

plt.axvline(0, color='black',linewidth=0.5)

plt.grid(True)

plt.legend()

plt.gca().set\_aspect('equal', adjustable='box')

plt.title("2D Transformations on a Rectangle")

plt.show()

**OUTPUT:**

