**PRACTICAL NO.4**

**Python program for Transformation of 2D Points by applying Scaling, Reflection, Rotation, Shearing and Translation**.

Roll No:- Name:-

**Example 1. Apply each of the following transformations on the point P = [4,3].**

**1. Reflection through Y-axpis.**

**2. Scaling in X-coordinate by factor 3.**

**3. Scaling in Y-coordinate by factor 9.2.**

**4. Reflection through the line y=2**

**5. Shearing in Y direction by 3 units.**

**6. Scaling in X and Y direction by and 2 units respectively.**

**7. Shearing in both X and Y direction by-3 and 1 units respectively.**

**8. Rotation about origin by an angle 45 degrees.**

**Code :-**

**from sympy import\***

**from numpy import\***

**p=Point(4,3)**

**q1=p.transform(Matrix([[-1,0,0],[0,1,0],[0,0,1]]))**

**print("Q1->",q1)**

**q2=p.transform(Matrix([[3,0,0],[0,1,0],[0,0,1]]))**

**print("Q2->",q2)**

**q3=p.transform(Matrix([[1,0,0],[0,3.2,0],[0,0,1]]))**

**print("Q3->",q3)**

**q4=p.transform(Matrix([[0,-1,0],[-1,0,0],[0,0,1]]))**

**print("Q4->",q4)**

**q5=p.transform(Matrix([[1,3,0],[0,1,0],[0,0,1]]))**

**print("Q5->",q5)**

**q6=p.transform(Matrix([[3/2,0,0],[0,2,0],[0,0,1]]))**

**print("Q6->",q6)**

**q7=p.transform(Matrix([[1,1,0],[-3,1,0],[0,0,1]]))**

**print("Q7->",q7)**

**q8=p.transform(Matrix([[round(cos(45\*pi/180),4),round(sin(45\*pi/180),4),0],[-round(sin(45\*pi/180),4),round(cos(45\*pi/180),4),0],[0,0,1]]))**

**print("Q8->",q8)**

**Output :-**

Q1-> Point2D(-4, 3)

Q2-> Point2D(12, 3)

Q3-> Point2D(4, 48/5)

Q4-> Point2D(-3, -4)

Q5-> Point2D(4, 15)

Q6-> Point2D(6, 6)

Q7-> Point2D(-5, 7)

Q8-> Point2D(7071/10000, 49497/10000)

**Example 2. Apply each of the following transformations on the point P = (-2,4).**

**1. Reflection through line 3x+4y=5.**

**2. Scaling in X-coordinate by factor 6.**

**3. Scaling in Y-coordinate by factor 4.1.**

**4. Reflection through the line y=2x+3.**

**5. Shearing in Y direction by 7 units.**

**6. Scaling in X and Y direction by and 4 units respectively.**

**7. Shearing in both X and Y direction by 4 and 7 units respectively.**

**8. Rotation about origin by an angle 48 degrees.**

**Code:-**

**from sympy import\***

**from numpy import\***

**p=Point(-2,4)**

**q1=p.transform(Matrix([[6,0,0],[0,1,0],[0,0,1]]))**

**print("Q1->",q1)**

**q2=p.transform(Matrix([[1,0,0],[0,4.1,0],[0,0,1]]))**

**print("Q2->",q2)**

**q3=p.transform(Matrix([[1,7,0],[0,1,0],[0,0,1]]))**

**print("Q3->",q3)**

**q4=p.transform(Matrix([[7/2,0,0],[0,4,0],[0,0,1]]))**

**print("Q4->",q4)**

**q5=p.transform(Matrix([[1,7,0],[4,1,0],[0,0,1]]))**

**print("Q5->",q5)**

**q8=p.transform(Matrix([[round(cos(48\*pi/180),4),-round(sin(48\*pi/180),4),0],[round(sin(48\*pi/180),4),round(cos(48\*pi/180),4),0],[0,0,1]]))**

**print("Q8->",q8)**

**Output :-**

Q1-> Point2D(-12, 4)

Q2-> Point2D(-2, 82/5)

Q3-> Point2D(-2, -10)

Q4-> Point2D(-7, 16)

Q5-> Point2D(14, -10)

Q8-> Point2D(8171/5000, 20813/5000)

**Example 3. Apply each of the following transformation on the point P [-2 3.5].**

**(a) Reflection through the line y = -z.**

**(b) Rotation about origin through an angle 57° in clockwise sense.**

**(c) Reflection through origin.**

**(d) Uniform scaling by factor 3.**

Code:-

**from sympy import\***

**from numpy import\***

**p=Point(-2,3.5)**

**q=p.transform(Matrix([[0,-1,0],[-1,0,0],[0,0,1]]))**

**print("Q1->",q)**

**x=57**

**q=p.transform(Matrix([[round(cos(x\*pi/180),4),round(sin(x\*pi/180),4),0],[round(sin(x\*pi/180),4),round(cos(x\*pi/180),4),0],[0,0,1]]))**

**print("Q2->",q)**

**q=p.transform(Matrix([[-1,0,0],[0,-1,0],[0,0,1]]))**

**print("Q3->",q)**

**q=p.transform(Matrix([[3,0,0],[0,3,0],[0,0,1]]))**

**print("Q4->",q)**

**Outpur :-**

Q1-> Point2D(-7/2, 2)

Q2-> Point2D(-80493/20000, 2287/10000)

Q3-> Point2D(2, -7/2)

Q4-> Point2D(-6, 21/2)

Example 4. Apply each of the following transformations on the point P [3-1].

(a) Reflection through X-axis.

(b) Scaling in X-coordinate by factor 2.

(c) Scaling in Y-coordinate by factor 1.5.

(d) Reflection through the line y = 1.

(e) Shearing in Y direction by 2 units.

(f) Scaling in X and Y direction by and 3 units respectively.

(g) Shearing in both X and Y direction by -2 and 4 units respectively.

(h) Rotation about origin by an angle 30 degrees.

Code:-

**from sympy import\***

**from numpy import\***

**p=Point(3,-1)**

**q1=p.transform(Matrix([[1,0,0],[0,-1,0],[0,0,1]]))**

**print("Q1->",q1)**

**q2=p.transform(Matrix([[2,0,0],[0,1,0],[0,0,1]]))**

**print("Q2->",q2)**

**q3=p.transform(Matrix([[1,0,0],[0,1.5,0],[0,0,1]]))**

**print("Q3->",q3)**

**q4=p.transform(Matrix([[0,1,0],[1,0,0],[0,0,1]]))**

**print("Q4->",q4)**

**q5=p.transform(Matrix([[1,2,0],[0,1,0],[0,0,1]]))**

**print("Q5->",q5)**

**q6=p.transform(Matrix([[1/2,0,0],[0,3,0],[0,0,1]]))**

**print("Q6->",q6)**

**q7=p.transform(Matrix([[1,4,0],[-2,1,0],[0,0,1]]))**

**print("Q7->",q7)**

**q8=p.transform(Matrix([[1,0,0],[0,1,0],[0,0,1]]))**

**print("Q8->",q8)**

**x=30**

**q9=p.transform(Matrix([[round(cos(x\*pi/180),4),round(sin(x\*pi/180),4),0],[round(sin(x\*pi/180),4),round(cos(x\*pi/180),4),0],[0,0,1]))**

**print("Q9->",q9)**

Output :-

Q1-> Point2D(3, 1)

Q2-> Point2D(6, -1)

Q3-> Point2D(3, -3/2)

Q4-> Point2D(-1, 3)

Q5-> Point2D(3, 5)

Q6-> Point2D(3/2, -3)

Q7-> Point2D(5, 11)

Q8-> Point2D(3, -1)

Q9-> Point2D(1049/500, 317/500)