Name – Ritesh Anil Badhe Roll No. 115

std -SY Bsc(CS)

Batch - F

Date 11/01/2025

Practical 3 & 4 : Application of Computational Geometry1) Apply Python program in each of the following transformations on the point P[3,−1]

In [ ]:
```
(I)     Refection through X-axis.
```

In [1]:
```
from sympy import*
P=Point(3,-1)
P.transform(Matrix([[1,0,0],[0,-1,0],[0,0,1]]))
```

Out[1]: Point2D(3,1)

(II)Scaling in X−coordinate by factor 2.

In [2]:
```
P.scale(2,0)
```

Out[2]: Point2D(6,0)

(III)Scaling in Y−coordinate by factor 1.5.

In [3]:
```
P.scale(0,1.5)
```

Out[3]: $\text{Point2D}\left(0,-\frac{3}{2}\right)$

(IV)Reflection through the line y = x.

In [5]:
```
P.scale(0,3/2)
```

Out[5]: $\text{Point2D}\left(0,-\frac{3}{2}\right)$

2) Apply Python program in each of the following transformations on the point P[3,8] (I) Refection through X−axis.

In [6]:
```
from sympy import*
P=Point(3,8)
P.transform(Matrix([[1,0,0],[0,-1,0],[0,0,1]]))
```

Out[6]: Point2D(3,−8)

(II)Scaling in X−coordinate by factor 6.

In [7]:
```
P.scale(6,0)
```

Out[7]: Point2D(18,0)

(III)Rotation about origin through an angle 30◦.

In [8]:
```
P.rotate(pi/6)
```

Out[8]: $\left(-4+\frac{\sqrt{3}}{2}, \frac{3}{2}+4\sqrt{3}\right)$ 3√3 3

Point2D

(IV)Reflection through the line y = −x.

In [9]:
```
x,y=symbols('x y')
P.reflect(Line(y+x))
```

Out[9]: Point2D(−8,−3)

3) Write a python program to apply the following transformations on the point (−2,4) :

In [ ]:
```
(I)Shearing in Y direction by 7 units.
```

In [10]:
```
from sympy import*
P=Point(-2,4)
P.transform(Matrix([[1,7,0],[0,1,0],[0,0,1]]))
```

Out[10]: Point2D(−2,−10)

(II)Scaling in X and Y direction by 7/2 and 7 units respectively.

In [12]:
```
P.scale(7/2,2)
```

Out[12]: Point2D(−7,8)

In [ ]:
```
(III)Shearing in X and Y direction by 4 and 7 units respectively.
```

In [13]:
```
P.transform(Matrix([[1,7,0],[4,1,0],[0,0,1]]))
```

Out[13]: Point2D(14,−10)

In [ ]:
```
(IV)Rotation about origin by an angle 60◦.
```

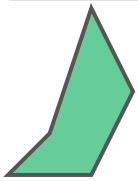
In [14]:
```
P.rotate(pi/3)
```

Out[14]: Point2D(−2√3 − 1,2 − √3)

4)Write a python program to draw polygon with vertices [3,3],[4,6],[5,4],[4,2] and [2,2], and its translation in x and y direction by factors -2 and 1 respectively.

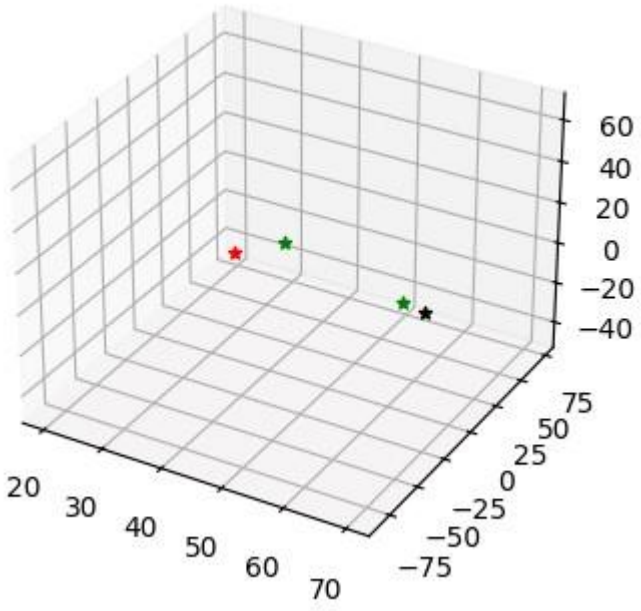
In [15]:
```
from sympy import*
A = Point(3,3)
B = Point(4,6)
C = Point(5,4)
D = Point(4,2)
E = Point(2,2)
P = Polygon(A,B,C,D,E)
P.translate(-2,1)
```

Out[15]:



5) Plot 3D axes with labels as x−axis and z−axis and also plot following points with given coordinates in one graph. (I) (70,−25,15) as a diamond in black colour, (II) (50,72,−45) as a ∗ in green colour, (III) (58,−82,65) as a dot in green colour, (IV) (20,72,−45) as a ∗ in Red colour.

In [26]:
```
from mpl_toolkits import mplot3d import
matplotlib.pyplot as plt import numpy as
np fig=plt.figure(figsize=(4,4))
ax=fig.add_subplot(111,projection='3d')
ax.scatter(70,-25,15,c='k',marker='*')
ax.scatter(50,72,-45,c='g',marker='*')
ax.scatter(58,-82,65,c='g',marker='*')
ax.scatter(20,72,-45,c='r',marker='*')
plt.show()
```



In [ ]:
```
6)      Find the combined transformation of the line segment between the points
(I)     Rotation about origin through an angle π.
(II)     Scaling in X- coordinate by 2 units.
(III)   Reflection through the line y = -x.
(IV)    Shearing in X direction by 4 units.
```

In [27]:
```
from sympy import*
A=Point(5,-2)
B=Point(4,3)
s=Segment (A,B)
s1=s.rotate(pi)
s1
```

Out[27]:

In [28]:
```
s2=s1.scale(2,0)
s2
```

Out[28]:

In [30]:
```
x,y=symbols('x,y')
s3=s2.reflect(Line(x+y))
s3
```

Out[30]:

In [32]:
```
Points=s3.points p=Points[0] q=Points[1]
p1=p.transform(Matrix([[1,0,0],[4,1,0],[0,0,1]]))
p1
```

Out[32]: Point2D(40,10) In

[33]:
```
q1=q.transform(Matrix([[1,0,0],[4,1,0],[0,0,1]])) q1
```

Out[33]: Point2D(32,8)

7) Find the combined transformation of the line segment between the points A[4,−1] & B[3,0] by using Python program for the following sequence of transformations: (I) Shearing in X direction by 9 units. (II)Rotation about origin through an angle π. (III)Scaling in X− coordinate by 2 units. (IV)Reflection through the line y = x.

In [38]:
```
from sympy import*
A=Point(4,-1) B=Point(3,0) s=Segment(A,B)
points=s.points p=points[0] q=points[1]
p1=p.transform(Matrix([[1,0,0],[9,1,0],[0,0,1]]))
q1=q.transform(Matrix([[1,0,0],[9,1,0],[0,0,1]]))
s=Segment(p1,q1) s1=s.rotate(pi) s2=s1.scale(2,0)
x,y=symbols('x,y') s3=s2.reflect(Line(y-x)) s3
```

Out[38]:

In [ ]:
```
```