① Generate a vector x in the interval [-7, 7] using numpy package with 50 subintervals

→

```
from pylab import *
import numpy as np
X = np.linspace(-7, 7, 50)
print(x).
```

② (-π, π) 50 subinterval.

→

```
a = np.linspace(-pi, pi, 50)
print (a).
```

③ Rotate the point A = (3, 4) by 90 degrees using python

→

```
from sympy import *
A = Point(3,4)
B = A.rotate(90)
print (B)
```

o/p:-

```
-4*sin(90) + 3*cos(90), 4*cos(90)+3*sin(90)
```

④ Apply scaling in x direction by 3 units on the point A=(3, 4)

→

```
from sympy import *
A = Point(3,4)
B = A.scale (3,1)
print(B)
```

o/p:-

```
Point2D(9,4)
```

5 Apply scaling in y direction by 3 units on pt A=(3,4)

→

```
from sympy import *
A = Point (3,4)
B= A. scale(1,3)
print (B)
```

o/p-

```
Point2D (3, 12)
```

6 Reflect point A=(3,4) through line x+y=0

→

```
from sympy import *
x,y = symbols('x y')
A = Point (3,4)
B = A. reflect (Line (x+y))
print (B)
```

o/p-

```
Point 2D(-4, -3)
```

7 Generate line passing through points (2,3) and (4,3) and find eq^n of line.

→

```
from sympy import *
x,y = symbols(' x y')
L1 = Line (Point (2,3), Point(4,3))
B = L1. equation ()
print(B)
```

o/p-

```
y-3
```

8 Generate line segment having endpoints (0,0) & (3,10) find length of line segment

→

```
from sympy import *
x,y = symbols ('x y')
```

```
S1 = Segment(Point(0,0) and Point(10, 10))
B = S1.length
print(B)
```
O/P:-
```
10*sqrt(2)
```

9 Generate line segment having endpoints (0,0) and (10, 10) find midpoint of line segment.

→

```
from sympy import *
x, y = symbols('x y')
S1 = Segment(Point(0,0), Point(10,10))
B = S1.midpoint
print(B)
```
o/p
```
Point2D(5,5)
```

10 Generate line passing through points (2,3) and (4,3) and find slope of the line

→

```
from sympy import *
x,y = symbols('x y')
L1 = Line(Point(2,3), Point(4,3))
B = L1. slope
print(B)
```
O/P:-
```
0
```

11 Rotate the ray by 90 degrees having starting point (0,0) in the direction of (4,4)

→

```
from sympy import *
~~x,~~ R1 = Ray(Point(0,0), Point(4,4))
B = R1. rotate(pi/2)
print(B)
```

o/P Ray 2D(Point2D(0,0) , Point2D(-4,4))

⑫ Rotate Line segment by 180 degrees having endpoints (1,0) and (2,-1)

→

```
from sympy import *
L1 = Line(Point(1,0), Point(2,-1))
B = L1 .rotate(pi)
print(B)
```

o/p. Line2D(Point2D(-1,0), Point2D(-2,1))

⑬ Generate polygon with vertices (0,0)(1,0)(2,2)(1,4) & find its area , ~~perime~~

→

```
from sympy import *
p1, p2, p3, p4 = [(0,0), (1,0), (2,2), (1,4)]
P = Polygon (p1, p2, p3, p4)
B = P.area
print (B)
```

o/p 4.

⑭ Generate polygon with vertices —11— perimeter

→

```
p1, p2, p3, p4 = [(0,0), (1,0), (2,2), (1,4)]
P = Polygon(p1, p2, p3, p4)
B = P.perimeter
print(B)
```

o/p. 1 + sqrt(17) + 2*sqrt(5)

⑮ —11— is convex

→

```
B = P.is_convex()
print (B)
```

o/p :- True

center ↓ radius ↓ sides ↓

Polygon((0,0), £5, n8)

(16) Generate regular polygon with center (0,0), radius 5 and 8 sides also find perimeter of polygon.

→

```
from sympy import *
P= Polygon ((0,0), 5, n=8)
B = P.perimeter
print(B)
```

o/p

```
40* sqrt(2 - sqrt(2))
```

(17) —11— find area.

→

```
from sympy import *
P= Polygon((0,0), 5, n=8)
B= P.area
print(B)
```

o/p:-

```
(400 - 200* sqrt(2)) / (-4+4* sqrt (2))
```

(18) Generate triangle with vertices (0,0), (4,0)(2,4) check whether triangle is isosceles triangle.

→

```
from sympy import *
t1, t2, t3 = [(0,0), (4,0), (2,4)]
T = Triangle (t1, t2, t3)
~~print(=~~
B = T. is_isosceles ()
print(B)
```

o/p.

```
True
```

```
B. is_right ()
B. is_scalene ()
```

(19) Generate triangle with sides 3, 4, & 5 units

→

```
from sympy import *
T= Triangle (sss = (3,4,5))
print(T)
```

o/p

```
Triangle (Point2D(0,0) , Point 2D(3,0) , Point(3,4))
```

(20) Generate Δ with sides 1, 2 & angle 30°.

→

```
from sympy import *
T= Triangle (sas = (1, 30, 2)
print(T)
```

o/p

```
Triangle (Point2D(0,0), Point 2D( 2,0) , Point 2D(sqrt (3)/2, ½))
```

(21) Generate Δ with vertices (0,0), (1, 0), (5,1).

→

```
from sympy import *
t1,t2,t3 = [(0,0) , (1,0) , (5,1)]
T= Triangle(t1, t2, t3)
print(T)
```

o/p

```
Triangle (Point2D(0,0), Point 2D(1,0), Point 2D(5,1))
```

(22) Rotate the Δ by 270°, having vertices (-1, 2) (2, -5) (-1, 7)

→

```
from sympy import *
t1,t2,t3 = [(-1,2), (2, -5), (-1,7)]
T= Triangle (t1, t2, t3)
B= T.rotate~~(2π)~~(3*pi/2)
print(B)
```

o/p.

```
Triangle (Point2D(2,1) , Point(-5,-2), Point2D(7,1))
```