Sahakar Maharshi Bhausaheb Santuji Thorat

College Sangamner

DEPARTMENT OF COMPUTER SCIENCE

Sub : Mathematics

Remark
Demonstrator's Signature Date:- / /20

Name:- Gorde Yash Somnath ___ Roll.No:- 21 Date:-__________________
Title of the expt:- Slip no 23 ___________ Page.no:-_____ Class:- BCS _________

Q1) Attempt any TWO of the following

A) Write a python program plot the graph of sin x and cos x in [0,π] in one figure with 2x1 subplots

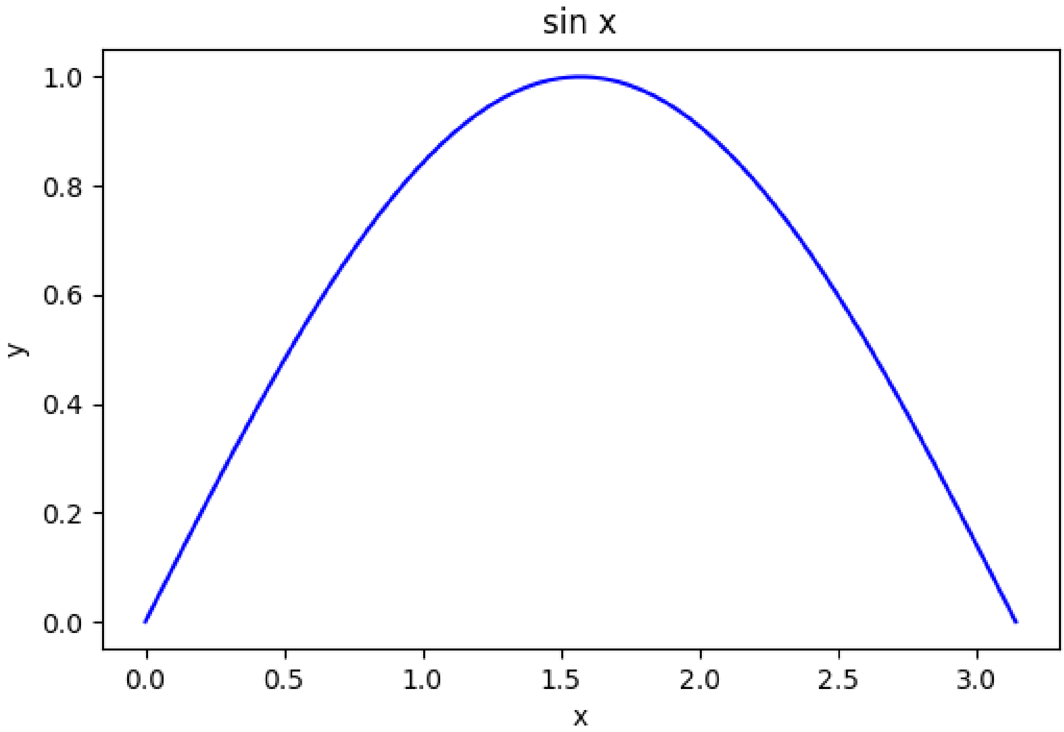
-→

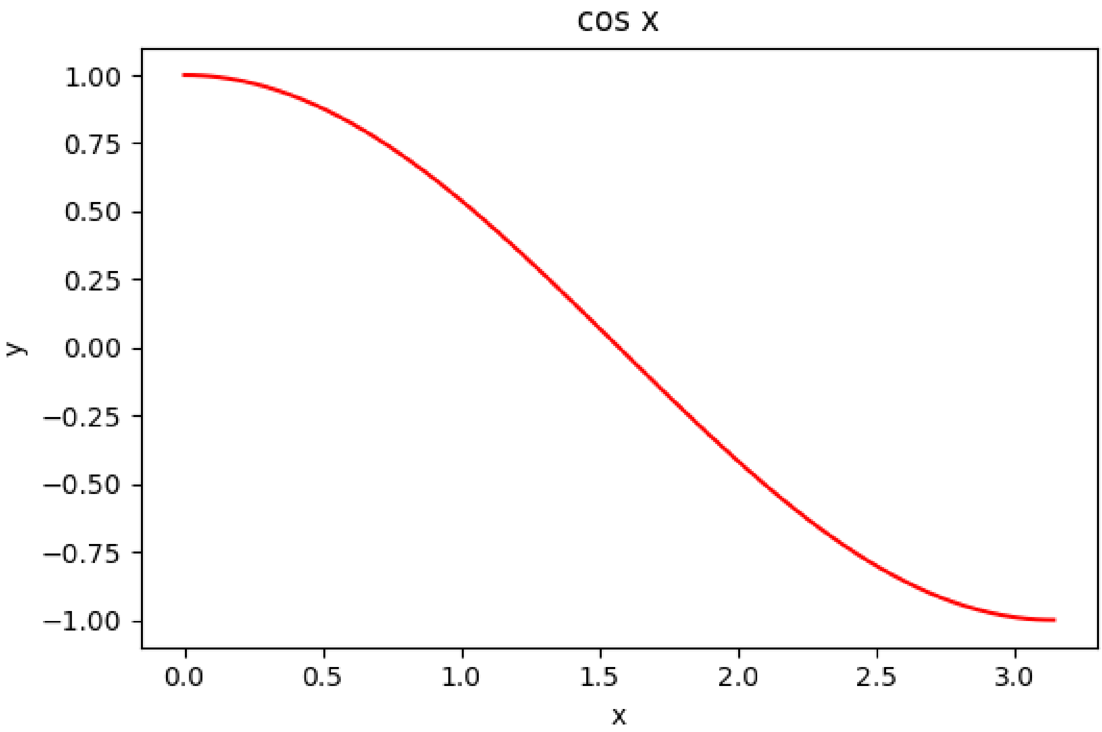
```
import numpy as np
import matplotlib.pyplot as plt
x = np.linspace(0, np.pi, 100)
y_sin = np.sin(x)
y_cos = np.cos(x)
fig, axs = plt.subplots(nrows=2, ncols=1, figsize=(6, 8))

axs[0].plot(x, y_sin, color='blue')
axs[0].set_title('sin x')
axs[0].set_xlabel('x')
axs[0].set_ylabel('y')

axs[1].plot(x, y_cos, color='red')
axs[1].set_title('cos x')
axs[1].set_xlabel('x')
axs[1].set_ylabel('y')

plt.tight_layout()
plt.show()
```





B) Write a python program to plot the graph of the function in the given interval

I) f(x)=x^3 in [0,5]

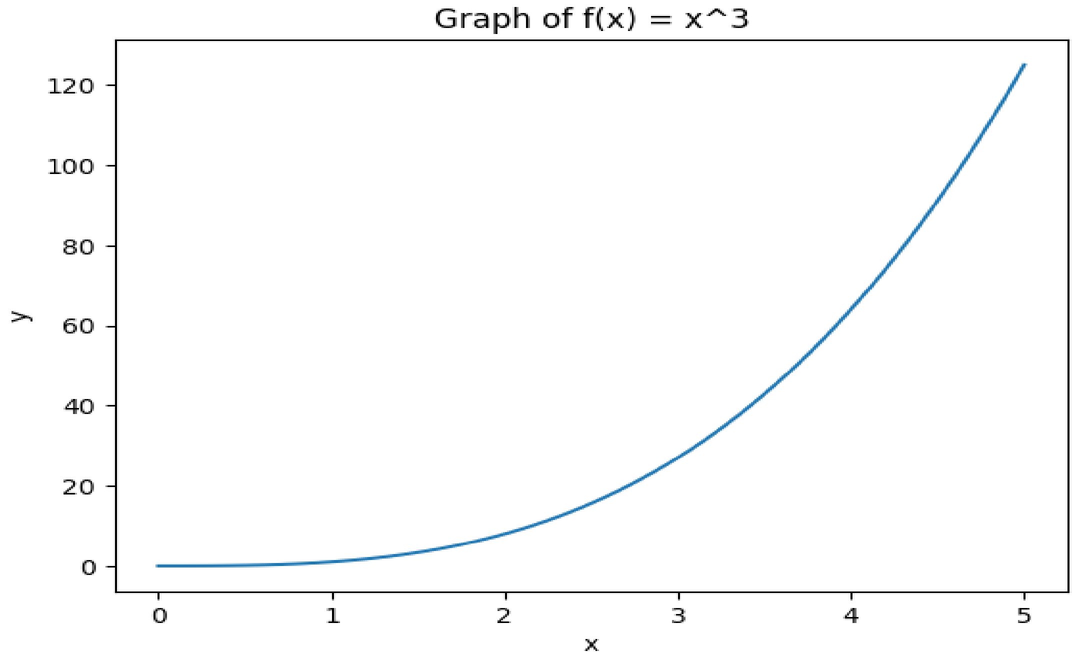
II) f(x)=x^2 in [-2,2]

-→

import numpy as np

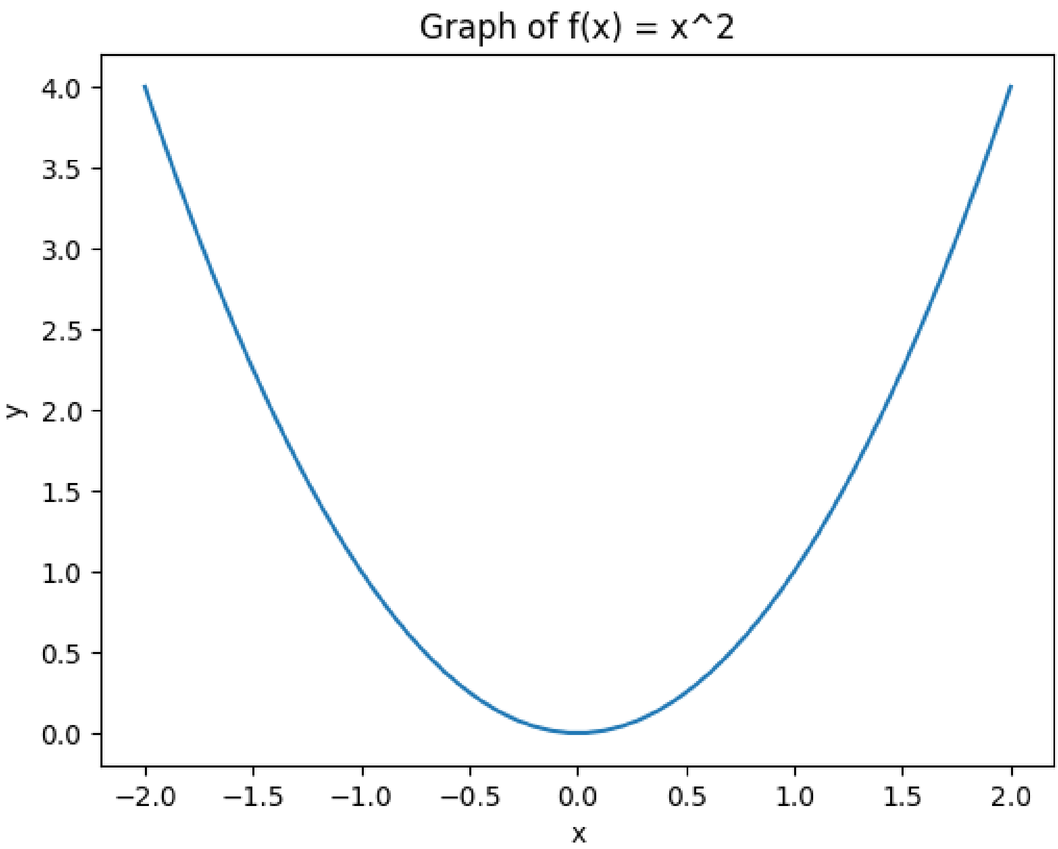
```
import matplotlib.pyplot as plt
def f(x):
    return x**3

x = np.linspace(0, 5)
plt.plot(x, f(x))
plt.title('Graph of f(x) = x^3')
plt.xlabel('x')
plt.ylabel('y')
plt.show()
```




```
import numpy as np
import matplotlib.pyplot as plt
def f(x):
    return x**2

x = np.linspace(-2, 2)
plt.plot(x, f(x))
plt.title('Graph of f(x) = x^2')
plt.xlabel('x')
plt.ylabel('y')
plt.show()
```



C) Write a python program to plot 3D surface plot of the function z=cos(|x|+|Y|) in -1<x,y<1

-→

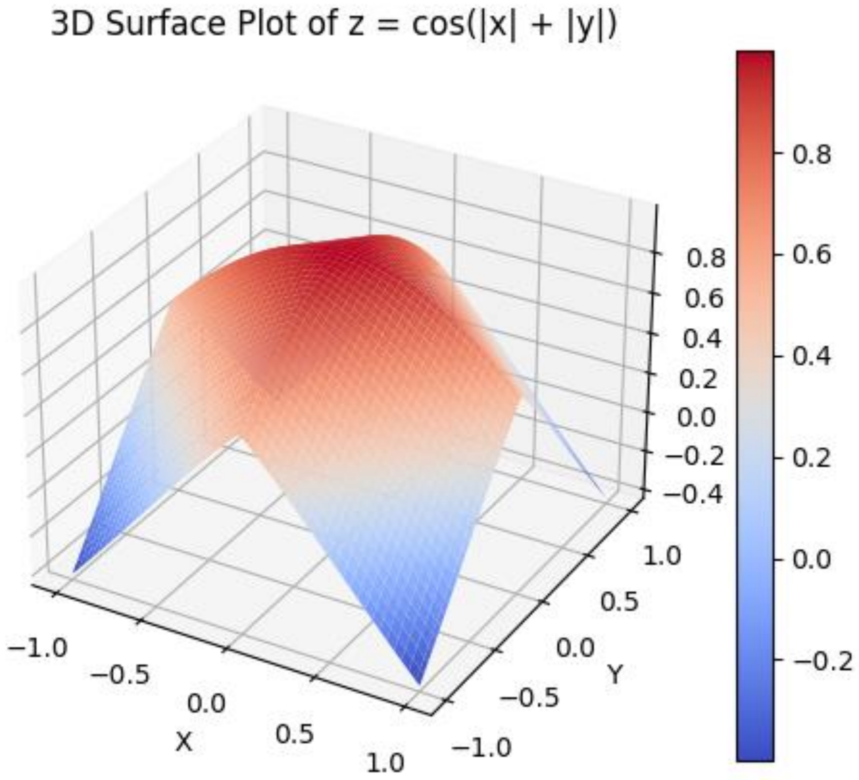
```
import matplotlib.pyplot as plt
import numpy as np
from mpl_toolkits.mplot3d import Axes3D
def f(x, y):
    return np.cos(np.abs(x) + np.abs(y))

x = np.linspace(-1, 1, 100)
y = np.linspace(-1, 1, 100)
X, Y = np.meshgrid(x, y)
Z = f(X, Y)

fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')
surf = ax.plot_surface(X, Y, Z, cmap='coolwarm')

ax.set_xlabel('X')
ax.set_ylabel('Y')
ax.set_zlabel('Z')
ax.set_title('3D Surface Plot of z = cos(|x| + |y|)')
```

```
fig.colorbar(surf)
plt.show()
```



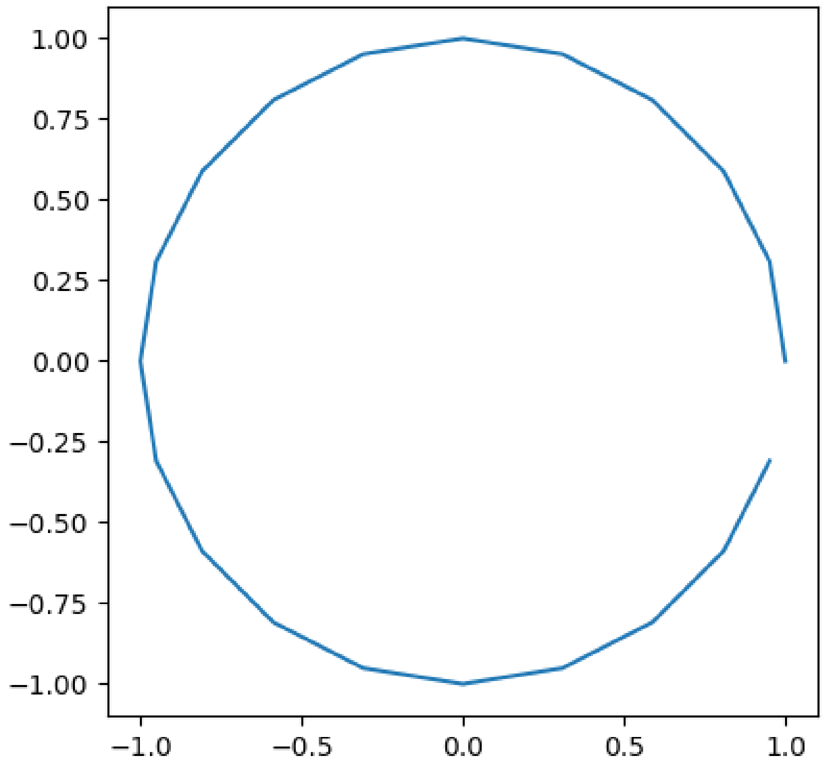
Q2) Attempt any TWO of the following

A) Write a python program to draw regular with 20 sides and radius 1 ceneterd at (0,0)
-→

```
import matplotlib.pyplot as plt
import numpy as np

theta = np.linspace(0, 2*np.pi, 20, endpoint=False)
x = np.cos(theta)
y = np.sin(theta)

fig, ax = plt.subplots()
ax.plot(x, y)
ax.set_aspect('equal')
plt.show()
```

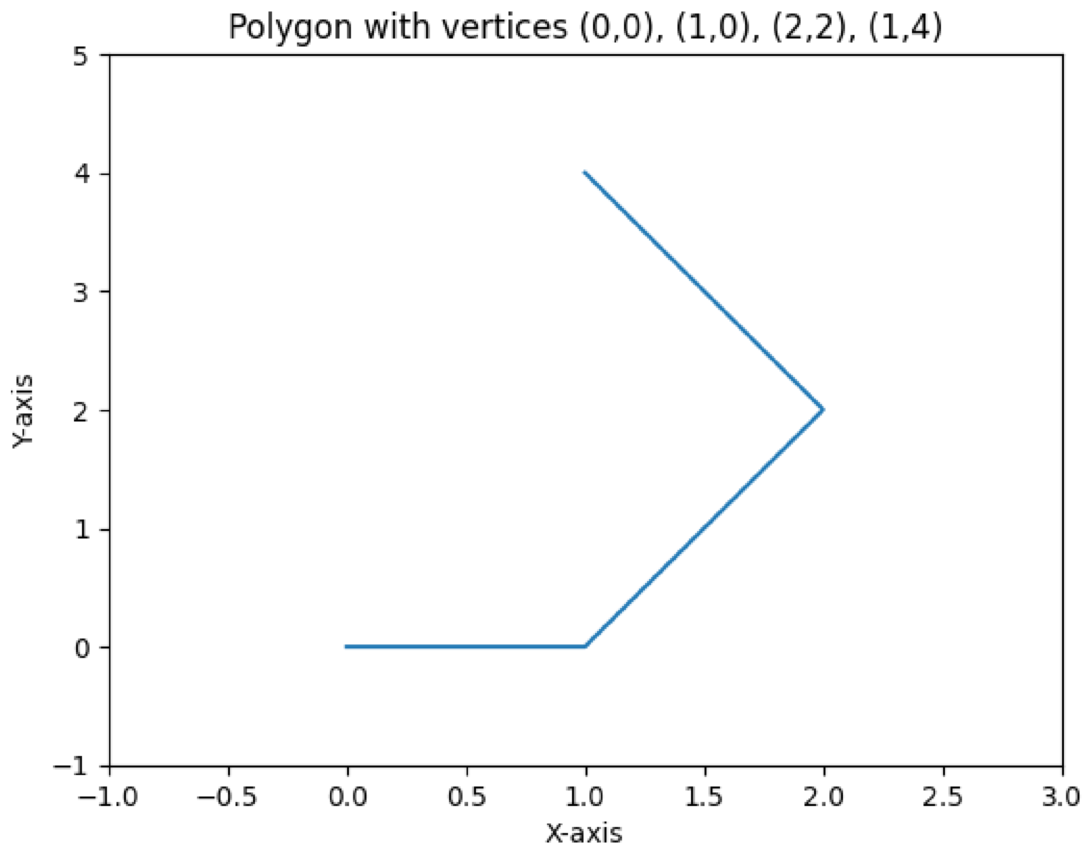


B) Write a python program to draw a polygon with vertices (0,0),(1,0),(2,2),(1,4), also find area of polygon

-→

```
import matplotlib.pyplot as plt
vertices = [(0,0), (1,0), (2,2), (1,4)]
x = [vertex[0] for vertex in vertices]
y = [vertex[1] for vertex in vertices]

plt.plot(x, y)
plt.xlim(-1, 3)
plt.ylim(-1, 5)
plt.xlabel('X-axis')
plt.ylabel('Y-axis')
plt.title('Polygon with vertices (0,0), (1,0), (2,2), (1,4)')
plt.show()
```



C)Write a python program to find area and perimeter of triangle ABC where A[0,1],B[-5,0] and C[-3,3]

-🡪

```
import math
A = [0, 1]
B = [-5, 0]
C = [-3, 3]
AB = math.sqrt((B[0]-A[0])**2 + (B[1]-A[1])**2)
BC = math.sqrt((C[0]-B[0])**2 + (C[1]-B[1])**2)
CA = math.sqrt((A[0]-C[0])**2 + (A[1]-C[1])**2)

perimeter = AB + BC + CA
s = perimeter / 2
area = math.sqrt(s * (s-AB) * (s-BC) * (s-CA))

print("Perimeter =", perimeter)
print("Area =", area)
```

output :

```
Perimeter = 12.310122064520764
Area = 6.500000000000002
```

Q3) Attempt the following

A) Attempt any ONE of the following

I) Write a python program to solve the following LPP :

Max Z=3x+5y+4z

Subjecct to 2x+3y≤8

2x+5y≤10

3x+2y+4z≤15

X,y,z≥0

-→

```
from scipy.optimize import linprog

c = [-3, -5, -4]
A = [[2, 3, 0], [2, 5, 0], [3, 2, 4]]
b = [8, 10, 15]
x0_bounds = (0, None)
x1_bounds = (0, None)
x2_bounds = (0, None)

res = linprog(c, A_ub=A, b_ub=b, bounds=[x0_bounds, x1_bounds, x2_bounds],
method='simplex')
print('Optimal value:', round(res.fun * -1, 2))
print('x:', res.x)
```

output :

```
Optimal value: 21.0
x: [0.   2.   2.75]
```

II) Write a python program to solve the following LPP :

Min Z=3x+5y+4z

Subject to 2x+2y≤12

2x+2y≤10

5x+2y≤10

X,y≥0

-→

```
from scipy.optimize import linprog
obj_coeff = [3, 5, 4]

lhs_ineq_coeff = [
   [2, 2, 0],
```

```
    [2, 2, 0],
    [5, 2, 0]
]

rhs_ineq_value = [12, 10, 10]
bounds = [(0, None), (0, None), (0, None)]
res = linprog(c=obj_coeff, A_ub=lhs_ineq_coeff, b_ub=rhs_ineq_value,
bounds=bounds, method='simplex')

print("Optimal value of Z:", round(res.fun, 2))
print("Optimal values of x, y, z:", res.x)
```

output :

```
Optimal value of Z: 0.0
Optimal values of x, y, z: [0. 0. 0.]
```

B) Attempt any ONE of the following

I) Write the python program to apply each of the following transformation on the point p[3,-1]

A) Reflection through X axis

B) Rotation about origin by an angle 30 degree

C) Scalling in Y coordinate by factor 8

D) Shering in X direction by 2 units

-→

```
import numpy as np
p = np.array([3, -1])

p_reflected = np.array([p[0], -p[1]])

angle = 30*np.pi/180
rotation_matrix = np.array([[np.cos(angle), -np.sin(angle)],
                            [np.sin(angle), np.cos(angle)]])
p_rotated = rotation_matrix @ p

scaling_matrix = np.array([[1, 0],
                           [0, 8]])
p_scaled = scaling_matrix @ p

shearing_matrix = np.array([[1, 0],
                            [2, 1]])
```

```
p_sheared = shearing_matrix @ p

# printing the results
print("Original point: ", p)
print("Reflection through X axis: ", p_reflected)
print("Rotation about origin by 30 degrees: ", p_rotated)
print("Scaling in Y coordinate by a factor of 8: ", p_scaled)
print("Shearing in X direction by 2 units: ", p_sheared)
```

output :

```
Original point:  [ 3 -1]
Reflection through X axis:  [3 1]
Rotation about origin by 30 degrees:  [3.09807621 0.6339746 ]
Scaling in Y coordinate by a factor of 8:  [ 3 -8]
Shearing in X direction by 2 units:  [3 5]
```

II) Write a python program to apply the each of the following transformation on the point P[-2,4]

A) Reflection through the line y=x+2
B) Scaling in Y-coordination by factor 2
C) Shering in X direction by units
D) Rotation about origin by an angle 60 degrees

-→

```
import math
P = [-2, 4]
m = 1
c = 2
x_new = (P[1] - c + m*P[0])/(1 + m**2)
y_new = (m*P[1] + m**2*P[0] + c)/(1 + m**2)
P_reflected = [x_new, y_new]
print("Reflection through the line y = x + 2:", P_reflected)

k = 2
x_new = P[0]
y_new = k*P[1]
P_scaled = [x_new, y_new]
print("Scaling in Y-coordinate by a factor of 2:", P_scaled)

k = 2
x_new = P[0] + k*P[1]
```

```
y_new = P[1]
P_sheared = [x_new, y_new]
print("Shearing in X-direction by 2 units:", P_sheared)

theta = math.radians(60)
x_new = P[0]*math.cos(theta) - P[1]*math.sin(theta)
y_new = P[0]*math.sin(theta) + P[1]*math.cos(theta)
P_rotated = [x_new, y_new]
print("Rotation about the origin by an angle of 60 degrees:", P_rotated)
```

output :

```
Reflection through the line y = x + 2: [0.0, 2.0]
Scaling in Y-coordinate by a factor of 2: [-2, 8]
Shearing in X-direction by 2 units: [6, 4]
Rotation about the origin by an angle of 60 degrees: [-4.464101615137754, 0.2679491924311 2325]
```