Sahakar Maharshi Bhausaheb Santuji Thorat

College Sangamner

DEPARTMENT OF COMPUTER SCIENCE

Sub : Mathematics

Remark
Demonstrator's Signature
Date:- / /20

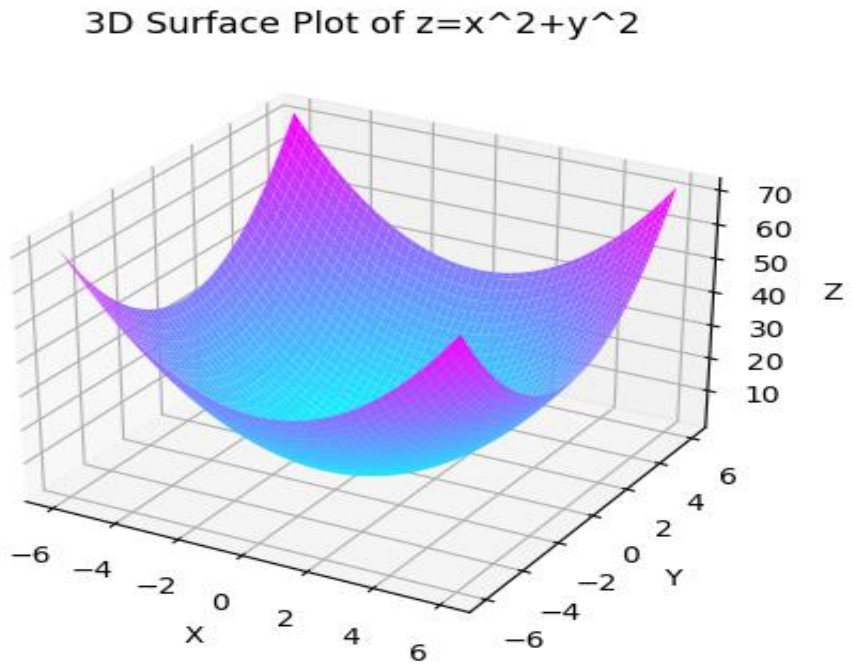
Name:-_Gorde Yash Somnath_____ Roll.No:-_21__Date:-_____
Title of the expt:-__Slip no 17_____Page.no:-_____Class:-_____BCS_____

Q1) Attempt any TWO of the following

A) Write a python program to plot the 3D graph of the function $z=x^2+y^2$ in -6<x,y<6 using surface plot

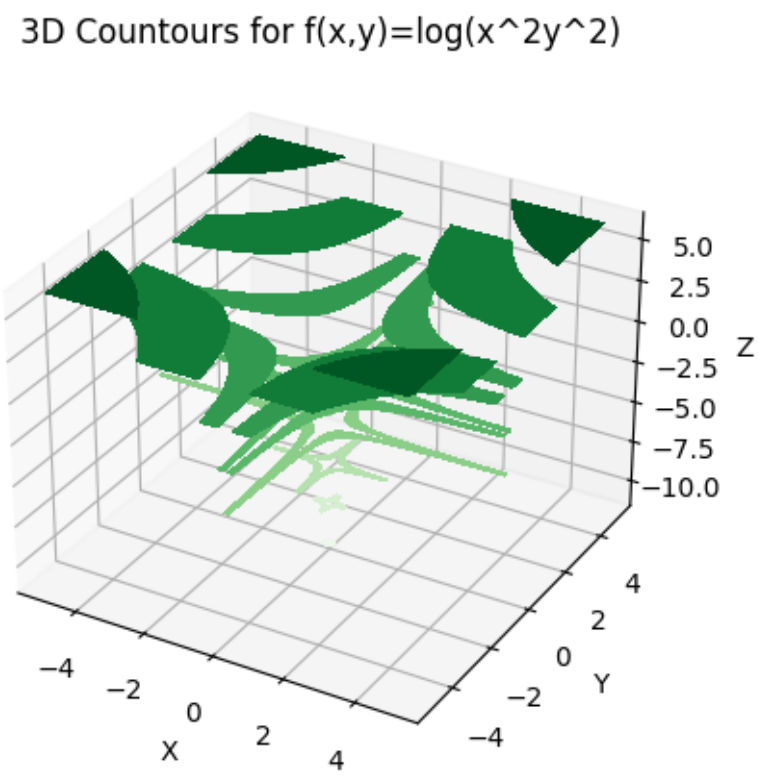
-→

```
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
import numpy as np
fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')
x = np.linspace(-6, 6, 100)
y = np.linspace(-6, 6, 100)
X, Y = np.meshgrid(x, y)
Z = X**2 + Y**2

ax.plot_surface(X, Y, Z, cmap='cool')
ax.set_xlabel('X')
ax.set_ylabel('Y')
ax.set_zlabel('Z')
ax.set_title('3D Surface Plot of z=x^2+y^2')
plt.show()
```



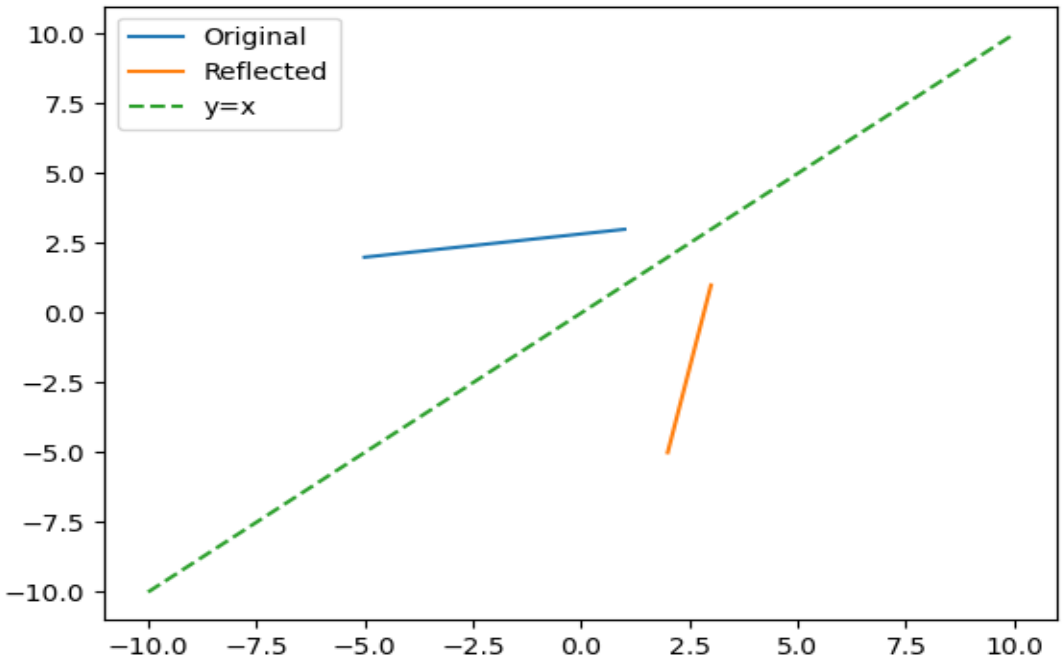
B) Write a python program to plot 3D countours for the function f(x,y)=log(x^2y^2) when -5≤x,y≤5 with greens color map
-→

```
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
def f(x, y):
    return np.log(x**2 * y**2)

x = np.linspace(-5, 5, 100)
y = np.linspace(-5, 5, 100)
X, Y = np.meshgrid(x, y)
Z = f(X, Y)
fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')

ax.contourf(X, Y, Z, cmap='Greens')
ax.set_xlabel('X')
ax.set_ylabel('Y')
ax.set_zlabel('Z')
ax.set_title('3D Countours for f(x,y)=log(x^2y^2)')
plt.show()
```



C) Write a python program to reflect the line segment joining the points A[-5,2] and B[1,3] through the line y=x
-→

```
A = [-5, 2]
B = [1, 3]

def y_equals_x(x):
    return x

def reflect_point(point):
    x, y = point
    return [y, x]

A_reflected = reflect_point(A)
B_reflected = reflect_point(B)

print("A_reflected:", A_reflected)
print("B_reflected:", B_reflected)

import matplotlib.pyplot as plt
plt.plot([A[0], B[0]], [A[1], B[1]], label="Original")
plt.plot([A_reflected[0], B_reflected[0]], [A_reflected[1], B_reflected[1]], label="Reflected")
plt.plot([-10, 10], [-10, 10], linestyle="dashed", label="y=x")
plt.legend()
plt.show()
```

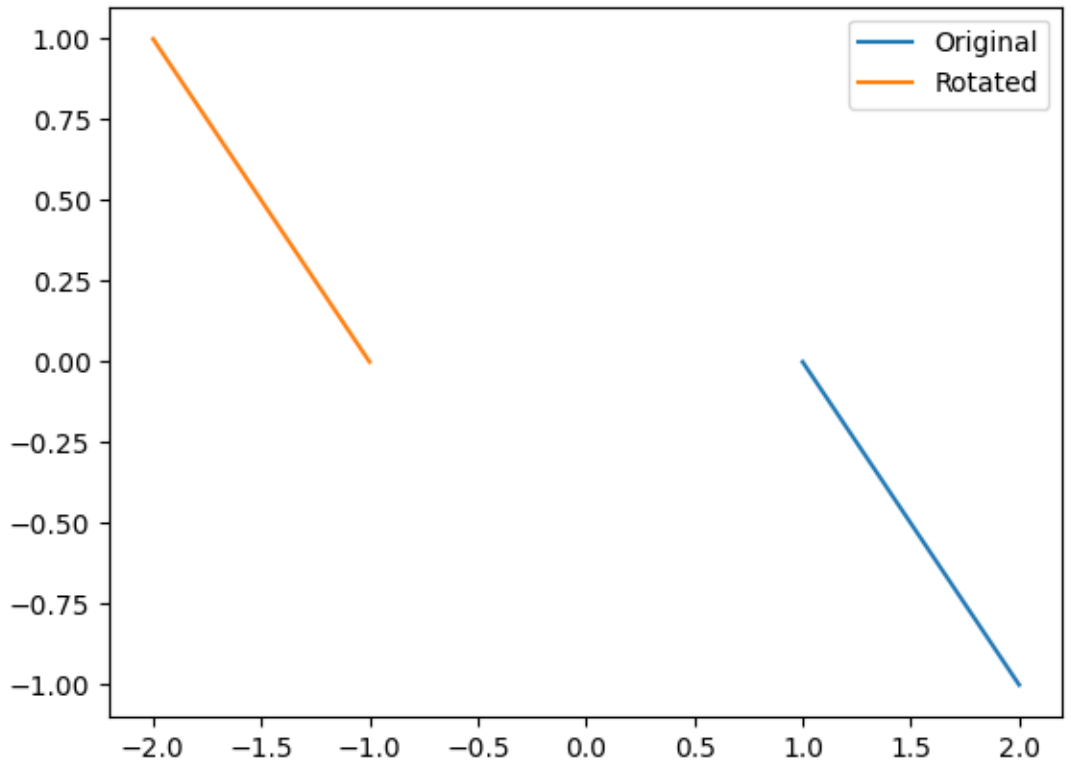


Q 2) Attempt any TWO of the following

A) Write a python program to rotate the line segment by 180 degrees having end points (1,0) and (2,-1)

-→

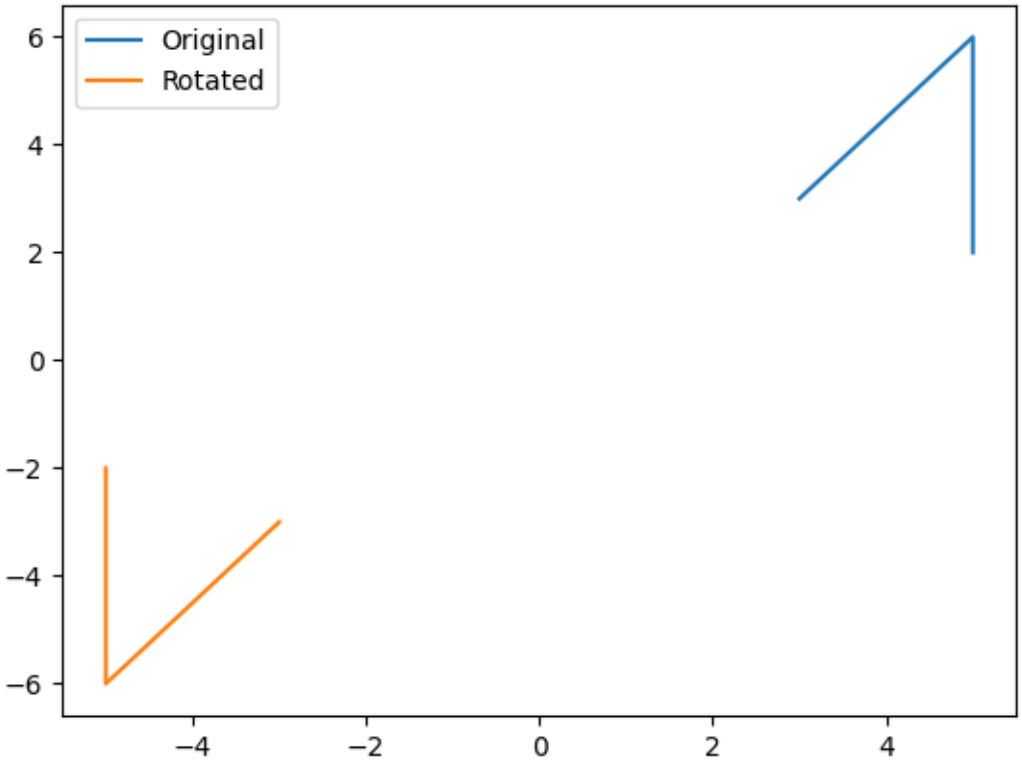
```
A = [1, 0]
B = [2, -1]

def rotate_point(point):
    x, y = point
    return [-x, -y]

A_rotated = rotate_point(A)
B_rotated = rotate_point(B)

print("A_rotated:", A_rotated)
print("B_rotated:", B_rotated)

import matplotlib.pyplot as plt
plt.plot([A[0], B[0]], [A[1], B[1]], label="Original")
plt.plot([A_rotated[0], B_rotated[0]], [A_rotated[1], B_rotated[1]], label="Rotated")
plt.legend()
plt.show()
```



B)Write a python program to plot triangle with vertices [3,3],[5,6],[5,2] and its rotation about the origin by angle –π radians

-→

```
import numpy as np
import matplotlib.pyplot as plt
vertices = np.array([[3, 3], [5, 6], [5, 2]])
theta = -np.pi
rotation_matrix = np.array([[np.cos(theta), -np.sin(theta)],
                            [np.sin(theta), np.cos(theta)]])
rotated_vertices = np.dot(vertices, rotation_matrix)

plt.plot(vertices[:, 0], vertices[:, 1], label="Original")
plt.plot(rotated_vertices[:, 0], rotated_vertices[:, 1], label="Rotated")
plt.legend()
plt.show()
```



C) Write a python program to drawn a polygon with vertices (0,0),(1,0),(2,2),(1,4) and find its area and perimeter

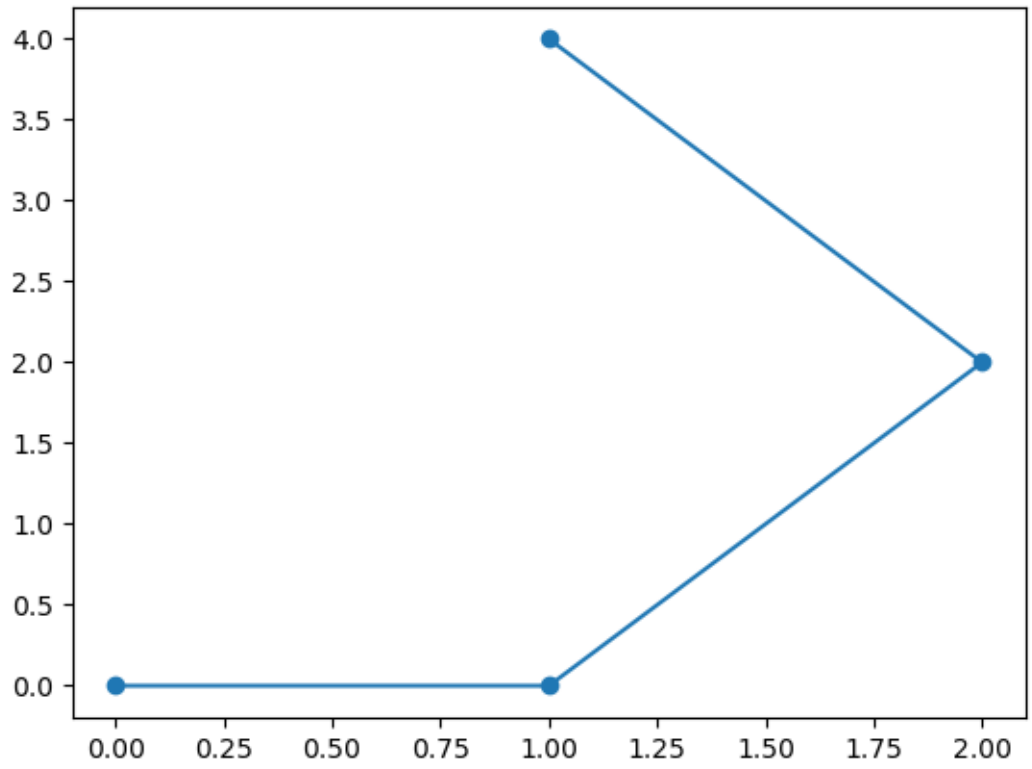
-→

```
import math
import matplotlib.pyplot as plt
vertices = [(0, 0), (1, 0), (2, 2), (1, 4)]

x, y = zip(*vertices)
plt.plot(x, y, '-o')
plt.show()
```

```
perimeter = 0
for i in range(len(vertices)):
    j = (i + 1) % len(vertices)
    dx = vertices[j][0] - vertices[i][0]
    dy = vertices[j][1] - vertices[i][1]
    perimeter += math.sqrt(dx**2 + dy**2)

area = 0.5 * abs(sum(x[i]*y[j] - y[i]*x[j] for i, j in zip(range(-1, len(x)-1), range(len(y)))))
print("Perimeter:", perimeter)
print("Area:", area)
```

output :

```
Perimeter: 9.595241580617241
Area: 4.0
```



Q3) Attempt the following

A) Attempt any one of the following

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

Max Z=4x+y+3z+5w

Subject to 4x+6y-5y-4w≥-20

-8x-3y+3z2w≤20

X,y≥0

-→

```
from scipy.optimize import linprog
obj = [-4, -1, -3, -5]
lhs_eq = [[4, 6, -5, -4], [-8, -3, 3, 2]]
rhs_eq = [20, -20]
bnd = [(0, None), (0, None), (0, None), (0, None)]
opt = linprog(c=obj, A_eq=lhs_eq, b_eq=rhs_eq, bounds=bnd, method="simplex")

print("Optimal solution:", opt.x)
print("Optimal objective value:", opt.fun)
```

output :

```
Optimal solution: [1.66666667 2.22222222 0.         0.        ]
Optimal objective value: -8.88888888888889
```

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

Max Z=x+y
Subject to x≤6
y≤6
x+y≤11
x,y≥0

-→

```
from scipy.optimize import linprog
obj = [-1, -1]
lhs_eq = [[1, 0], [0, 1], [1, 1]]
rhs_eq = [6, 6, 11]
bnd = [(0, None), (0, None)]
opt = linprog(c=obj, A_eq=lhs_eq, b_eq=rhs_eq, bounds=bnd, method="simplex")

print("Optimal solution:", opt.x)
print("Optimal objective value:", -opt.fun)
```

output :

```
Optimal solution: [0. 0.]
Optimal objective value: -0.0
```

B)Attempt any ONE of the following

I)Apply each of the following transformation on the point P[3,-1]

a) **Reflection through x-axis**
b) **Scaling in Y-coordinate by factor 1.5**
c) **Shering in both X and Y direction by -2 and 4 units respectively**
d) **Rotation about origin by anagle 30 degree**

-→

```
import math
P = [3, -1]

Px = P[0]
Py = -P[1]
print("Reflection through x-axis:", [Px, Py])

Px = P[0]
Py = 1.5 * P[1]
print("Scaling in Y-coordinate by factor 1.5:", [Px, Py])

Px = P[0] - 2 * P[1]
Py = P[1] + 4 * P[0]
print("Shearing in both X and Y direction by -2 and 4 units respectively:", [Px, Py])

theta = math.radians(30)
Px = P[0] * math.cos(theta) - P[1] * math.sin(theta)
Py = P[0] * math.sin(theta) + P[1] * math.cos(theta)
print("Rotation about origin by anagle 30 degree:", [Px, Py])
```

output :

```
Reflection through x-axis: [3, 1]
Scaling in Y-coordinate by factor 1.5: [3, -1.5]
Shearing in both X and Y direction by -2 and 4 units respectively: [5, 11]
Rotation about origin by anagle 30 degree: [3.098076211353316, 0.6339745962155611]
```

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

-→

```
import matplotlib.pyplot as plt
polygon_vertices = [[3,3], [4,6], [5,4], [4,2], [2,2]]
x, y = zip(*polygon_vertices)
plt.plot(x, y, marker='o', color='blue')
plt.title('Original Polygon')
plt.show()

x_factor = -2
```

```
y_factor = 1

transformed_vertices = []
for vertex in polygon_vertices:
    transformed_vertex = [vertex[0] * x_factor, vertex[1] * y_factor]
    transformed_vertices.append(transformed_vertex)

x, y = zip(*transformed_vertices)
plt.plot(x, y, marker='o', color='green')
plt.title('Transformed Polygon')
plt.show()
```