**PRACTICAL-12**

**AIM:** Write a program to implement all the functionalities of the MatPlotLib, NetworkX library in Python.

**Source Code:**

import networkx as nx

import matplotlib.pyplot as plt

import numpy as np

G = nx.Graph()

def display\_graph(G):

pos = nx.spring\_layout(G)

nx.draw(G, pos, with\_labels=True, node\_size=500, node\_color='lightblue', font\_size=12)

plt.title("Network Graph")

plt.axis('off')

plt.show()

G.add\_node(1)

display\_graph(G) # Display the graph after adding a node

G.add\_nodes\_from([2, 3, 4, 7, 9])

G.add\_edges\_from([(1, 2), (3, 1), (2, 4), (4, 1), (9, 1), (1, 7), (2, 9)])

display\_graph(G) # Display the graph after adding nodes and edges

G.remove\_node(3)

display\_graph(G) # Display the graph after removing a node

G.remove\_edge(1, 2)

display\_graph(G) # Display the graph after removing an edge

n = G.number\_of\_nodes()

m = G.number\_of\_edges()

print("Number of nodes:", n)

print("Number of edges:", m)

d = G.degree(2)

print("Degree of node 2:", d)

neighbor\_list = list(G.neighbors(2))

print("Neighbors of node 2:", neighbor\_list)

G.clear()

data = [25, 30, 15, 10, 20]

labels = ['Category A', 'Category B', 'Category C', 'Category D', 'Category E']

plt.figure(1)

plt.pie(data, labels=labels, autopct='%1.1f%%')

plt.title('Pie Chart')

plt.figure(2)

plt.bar(labels, data)

plt.title('Bar Chart')

x = np.arange(0, 10, 0.1)

y = np.sin(x)

plt.figure(3)

plt.plot(x, y)

plt.title('Line Plot')

x = np.random.rand(50)

y = np.random.rand(50)

plt.figure(4)

plt.scatter(x, y)

plt.title('Scatter Plot')

data = np.random.normal(0, 1, 1000)

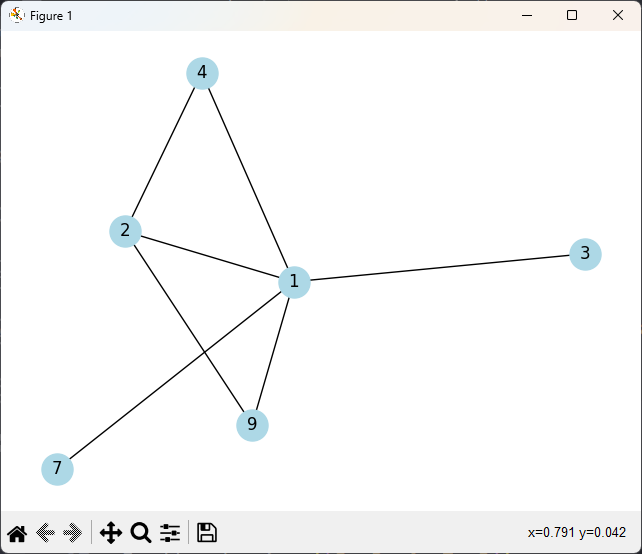
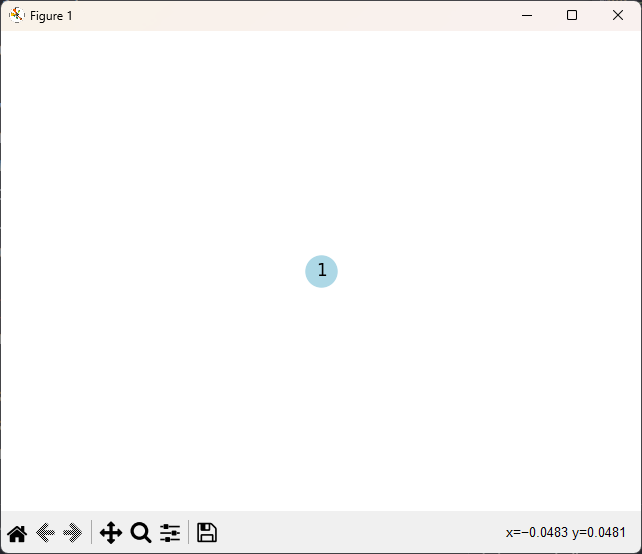
plt.figure(5)

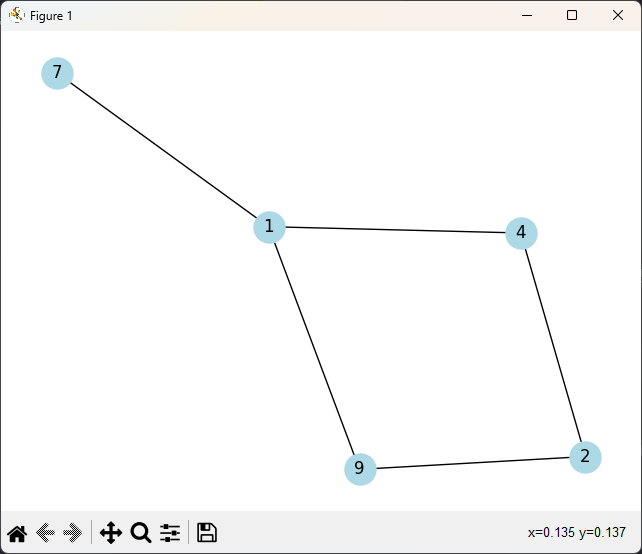
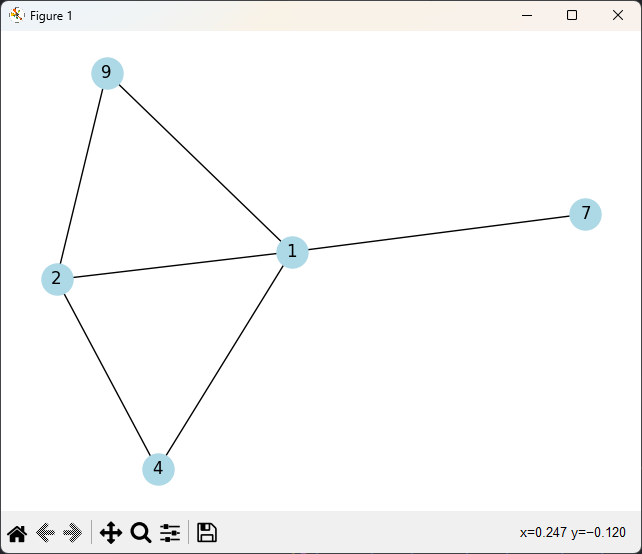
plt.hist(data, bins=30)

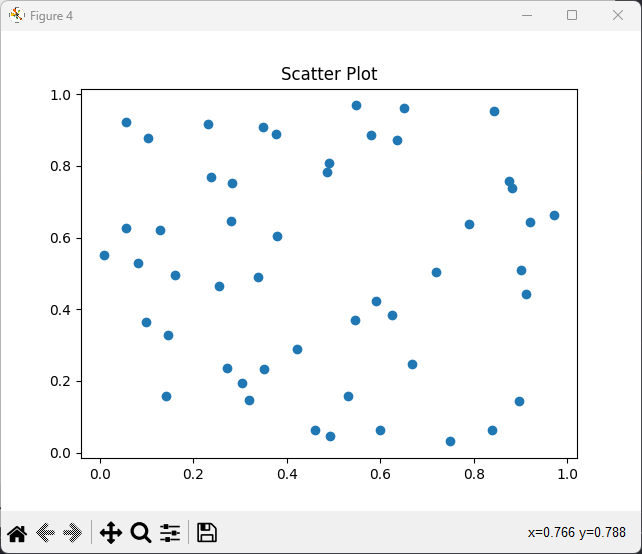
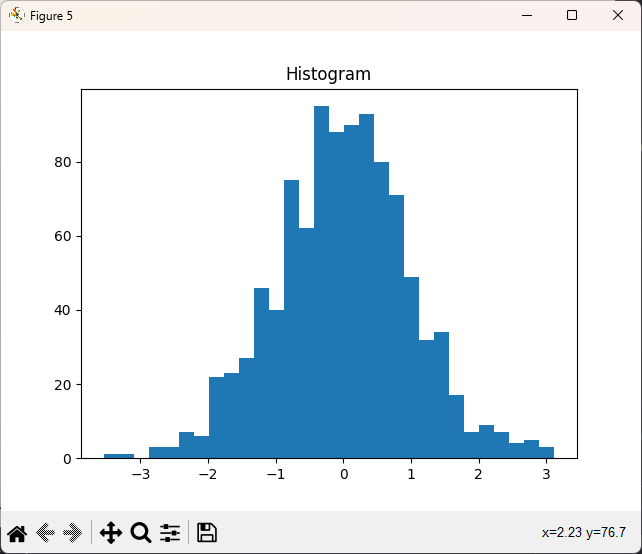
plt.title('Histogram')

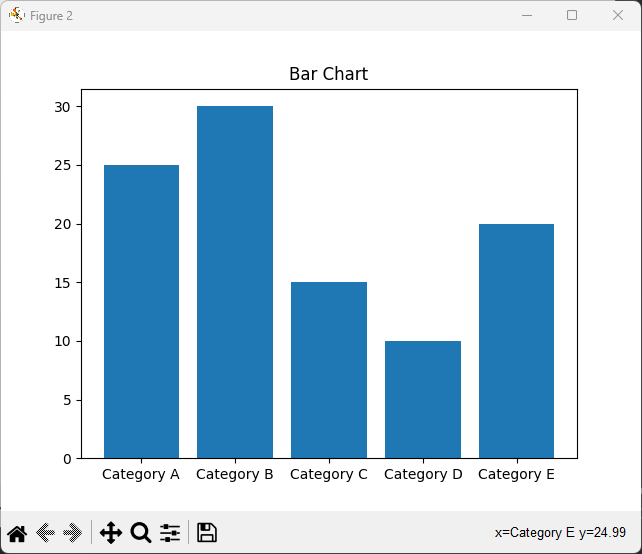
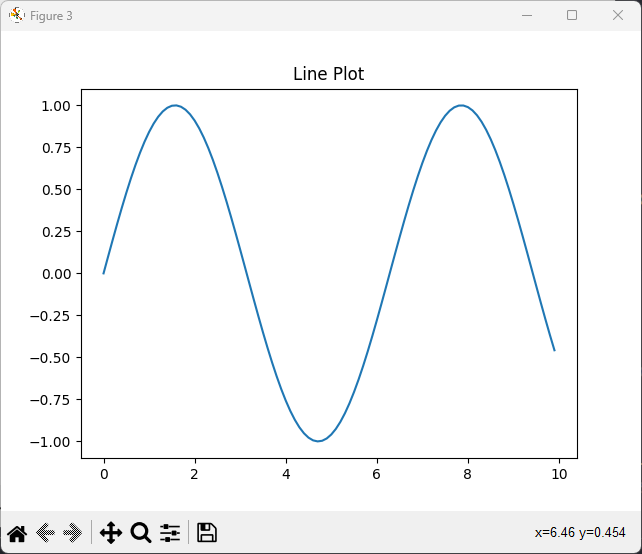
plt.show()

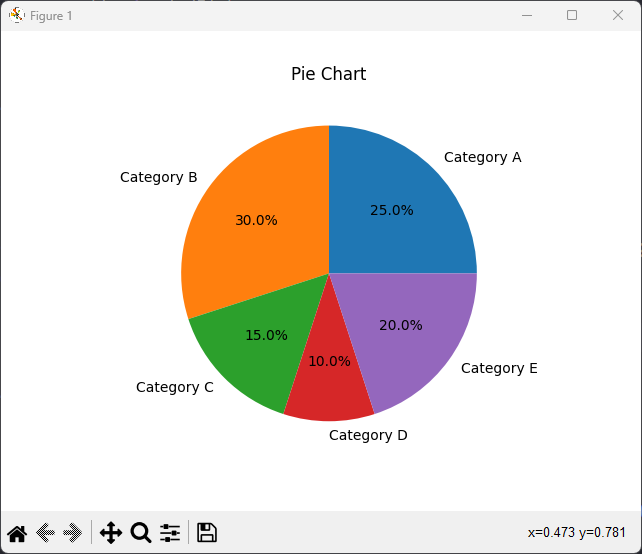
**Output:**









­­­­­­­­