NAME : PAREENITA A.SHIRSATH B.E.A.I.&.D.S. ROLL.NO : 57 PRN : 221101062

**BDA EXPERIMENT NO : 05**

**CODE:**

import numpy as np

import matplotlib.pyplot as plt

from scipy.spatial.distance import euclidean

def find\_similar\_items\_and\_plot(data\_points, threshold):

    """

    Finds similar items based on Euclidean distance and plots them.

    Args:

        data\_points (dict): A dictionary where keys are item names and values

                            are lists/tuples representing their coordinates.

        threshold (float): The maximum Euclidean distance for items to be

                           considered similar.

    """

    item\_names = list(data\_points.keys())

    num\_items = len(item\_names)

    # Calculate distances and identify similar pairs

    similar\_pairs = []

    for i in range(num\_items):

        for j in range(i + 1, num\_items):

            item1\_name = item\_names[i]

            item2\_name = item\_names[j]

            point1 = data\_points[item1\_name]

            point2 = data\_points[item2\_name]

            distance = euclidean(point1, point2)

            if distance <= threshold:

                similar\_pairs.append((item1\_name, item2\_name, distance))

    print(f"Similar pairs found (distance <= {threshold}):")

    for item1, item2, dist in similar\_pairs:

        print(f"  {item1} - {item2}: {dist:.2f}")

    # Plotting the data points and connections

    plt.figure(figsize=(10, 8))

    for name, coords in data\_points.items():

        plt.scatter(coords[0], coords[1], label=name, s=100) # s is marker size

        plt.text(coords[0] + 0.1, coords[1] + 0.1, name) # Labeling points

    # Draw lines for similar pairs

    for item1, item2, \_ in similar\_pairs:

        point1 = data\_points[item1]

        point2 = data\_points[item2]

        plt.plot([point1[0], point2[0]], [point1[1], point2[1]], 'r--', alpha=0.6) # Red dashed line

    plt.title("Item Similarity based on Euclidean Distance")

    plt.xlabel("Dimension 1")

    plt.ylabel("Dimension 2")

    plt.grid(True)

    plt.legend()

    plt.show()

# Example

if \_\_name\_\_ == "\_\_main\_\_":

    # Sample data points (e.g., features of items)

    items\_data = {

        "Maharashtra": (19.75, 75.7),

        "Delhi": (28.70, 77.10),

        "Punjab": (31.14, 75.34),

        "Kerala": (10.8, 76.27),

        "Haryana": (29.05, 76.08),

        "Gujrat": (22.25, 371.19),

        "Rajasthan": (27.02, 74.21),

    similarity\_threshold = 20.0  # Define the maximum distance for similarity

    find\_similar\_items\_and\_plot(items\_data, similarity\_threshold)

**OUTPUT:**



