1. You are given a small network of 4 cities connected by roads with the following distances:

City 1 to City 2: 3

City 1 to City 3: 8

City 1 to City 4: -4

City 2 to City 4: 1

City 2 to City 3: 4

City 3 to City 1: 2

City 4 to City 3: -5

City 4 to City 2: 6

Implement Floyd's Algorithm to find the shortest path between all pairs of cities. Display the distance matrix before and after applying the algorithm. Identify and print the shortest path from City 1 to City 3.

Program:’

# Number of cities

n = 4

# Initialize the distance matrix with large values (representing infinity)

inf = float('inf')

dist = [

[0, 3, 8, -4],

[inf, 0, 4, 1],

[2, inf, 0, inf],

[inf, 6, -5, 0]

]

# Display the distance matrix before applying the algorithm

print("Distance matrix before applying Floyd-Warshall algorithm:")

for row in dist:

print(row)

# Implementing Floyd-Warshall Algorithm

for k in range(n):

for i in range(n):

for j in range(n):

if dist[i][j] > dist[i][k] + dist[k][j]:

dist[i][j] = dist[i][k] + dist[k][j]

# Display the distance matrix after applying the algorithm

print("\nDistance matrix after applying Floyd-Warshall algorithm:")

for row in dist:

print(row)

# Identify the shortest path from City 1 to City 3

city\_1\_to\_3\_distance = dist[0][2]

print(f"\nThe shortest path from City 1 to City 3 is: {city\_1\_to\_3\_distance}")

2. Consider a network with 6 routers. The initial routing table is as follows:

Router A to Router B: 1

Router A to Router C: 5

Router B to Router C: 2

Router B to Router D: 1

Router C to Router E: 3

Router D to Router E: 1

Router D to Router F: 6

Router E to Router F: 2

Program:’

# Number of routers

n = 6

# Initialize the distance matrix with large values (representing infinity)

inf = float('inf')

dist = [

[0, 1, 5, inf, inf, inf], # Distances from Router A

[inf, 0, 2, 1, inf, inf], # Distances from Router B

[inf, inf, 0, inf, 3, inf], # Distances from Router C

[inf, inf, inf, 0, 1, 6], # Distances from Router D

[inf, inf, inf, inf, 0, 2], # Distances from Router E

[inf, inf, inf, inf, inf, 0] # Distances from Router F

]

# Display the distance matrix before applying the algorithm

print("Distance matrix before applying Floyd-Warshall algorithm:")

for row in dist:

print(row)

# Implementing Floyd-Warshall Algorithm

for k in range(n):

for i in range(n):

for j in range(n):

dist[i][j] = min(dist[i][j], dist[i][k] + dist[k][j])

# Display the distance matrix after applying the algorithm

print("\nDistance matrix after applying Floyd-Warshall algorithm:")

for row in dist:

print(row)

# Identify the shortest path from each router to every other router

print("\nShortest paths between each pair of routers:")

for i in range(n):

for j in range(n):

if i != j:

print(f"The shortest path from Router {chr(65 + i)} to Router {chr(65 + j)} is: {dist[i][j]}")