1. **Floyd’s algorithm**

**Code:**

#include <stdio.h>

#include <limits.h>

#define V 10 // Maximum number of vertices in the graph

// Function to print the distance matrix

void printSolution(int dist[][V], int n) {

printf("Shortest distances between every pair of vertices:\n");

for (int i = 0; i < n; i++) {

for (int j = 0; j < n; j++) {

if (dist[i][j] == INT\_MAX)

printf("INF\t");

else

printf("%d\t", dist[i][j]);

}

printf("\n");

}

}

// Function to perform Floyd's algorithm

void floydWarshall(int graph[][V], int n) {

int dist[V][V]; // Output matrix that will have the shortest distances between every pair of vertices

// Initialize distance matrix

for (int i = 0; i < n; i++)

for (int j = 0; j < n; j++)

dist[i][j] = graph[i][j];

// Update distance matrix by considering all vertices as intermediate vertices

for (int k = 0; k < n; k++) {

for (int i = 0; i < n; i++) {

for (int j = 0; j < n; j++) {

// If vertex k is on the shortest path from i to j, then update the value of dist[i][j]

if (dist[i][k] != INT\_MAX && dist[k][j] != INT\_MAX && dist[i][k] + dist[k][j] < dist[i][j])

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

}

}

}

// Print the shortest distances

printSolution(dist, n);

}

int main() {

int n;

// Input the number of vertices in the graph

printf("Enter the number of vertices in the graph (maximum %d): ", V);

scanf("%d", &n);

// Input the adjacency matrix representing the graph

int graph[V][V];

printf("Enter the adjacency matrix representing the graph (INF for no edge):\n");

for (int i = 0; i < n; i++) {

for (int j = 0; j < n; j++) {

scanf("%d", &graph[i][j]);

}

}

// Run Floyd's algorithm

floydWarshall(graph, n);

return 0;

}

**Output:**

Enter the number of vertices in the graph (maximum 10): 5

Enter the adjacency matrix representing the graph (INF for no edge):

1 2 3 4 5

0 1 2 3 4

2 3 4 5 6

3 4 5 6 7

4 5 6 7 8

Shortest distances between every pair of vertices:

1 2 3 4 5

0 1 2 3 4

2 3 4 5 6

3 4 5 6 7

4 5 6 7 8

--------------------------------

Process exited after 27.09 seconds with return value 0

Press any key to continue . . .

