Experiment No.8

Name :-Shivani Kolekar

Roll No:-24141031

Batch:-I2

Title:- All pair shortest path Problem using Floyd's algorithm.

Programm:-

```
#include <stdio.h>
#define INF 99999
#define V 4   // Number of vertices
void floydWarshall(int graph[V][V]) {
   int dist[V][V];
   // Initialize distance matrix same as input graph
   for (int i = 0; i < V; i++)
      for (int j = 0; j < V; j++)
         dist[i][j] = graph[i][j];
   // Floyd-Warshall algorithm (Dynamic Programming)
   for (int k = 0; k < V; k++) {
      for (int i = 0; i < V; i++) {
         for (int j = 0; j < V; j++) {
            if (dist[i][k] + dist[k][j] < dist[i][j])
               dist[i][j] = dist[i][k] + dist[k][j];
         }
      }
   }
   // Print the shortest distance matrix
   printf("\nAll Pairs Shortest Path Matrix:\n");
```

```
    for (int i = 0; i < V; i++) {
        for (int j = 0; j < V; j++) {
            if (dist[i][j] == INF)
                printf("%7s", "INF");
            else
                printf("%7d", dist[i][j]);
        }
        printf("\n");
    }
}
int main() {
    int graph[V][V] = {
        {0,   5,  INF, 10},
        {INF, 0,   3, INF},
        {INF, INF, 0,   1},
        {INF, INF, INF, 0}
    };
    printf("Floyd's Algorithm - All Pair Shortest Path\n");
    floydWarshall(graph);
    return 0;
}
```

Output:-

Complexities of Floyd's Algorithm (in short):

- **Time Complexity:** O(V3)O(V^3)O(V3)
  → Three nested loops for all pairs and intermediate vertices.
- **SpaceComplexity:** O(V2)O(V^2)O(V2)
  → Stores a distance matrix of size V×VV \times VV×V.

Applications of Floyd's Algorithm

1. Network Routing

- Used to find the shortest communication paths between all nodes in a network.
- Example: Internet routing protocols, packet switching networks.

---

2. Transportation and Navigation Systems

- Helps determine shortest travel routes between all cities/intersections.
- Used in:
  - GPS navigation systems
  - Airline and railway route planning

---

3. Social Network Analysis

- To find the degrees of separation or closeness between users (nodes).
- Example: "Friend-of-a-friend" connections on social platforms.

4. Urban Traffic Management

- Used to plan and optimize traffic flow by computing shortest paths between all junctions in a city map.

---

5. Game Development and AI Pathfinding

- Determines minimum movement costs between all positions or waypoints on a game map.

---

6. Telecommunication Systems

- Used in network optimization — finding least-cost communication paths between switches or routers.

---

7. Project Scheduling

- Helps analyze critical paths or shortest completion times in task dependency graphs.

Conclusion:-Floyd's Algorithm efficiently finds the shortest paths between all pairs of vertices using Dynamic Programming, with a time complexity of $O(V^3)$ — simple and powerful for small or medium-sized graphs.