# Experiment No.8

## All-Pairs Shortest Paths Problem using Floyd's Algorithm

## Reg.No.24141028

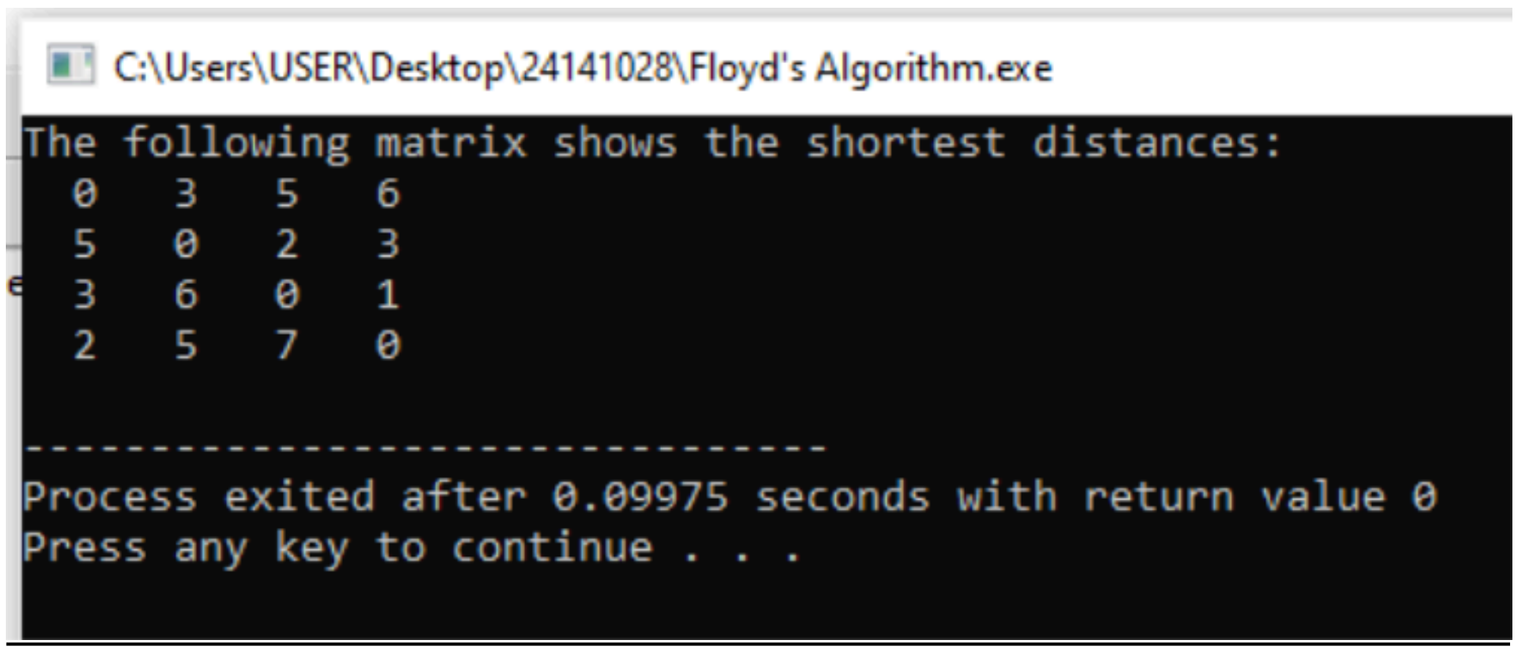
## Program :

```
#include <stdio.h>

#define V 4

#define INF 9999      // a large number representing infinity

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

        printf("The following matrix shows the shortest distances:\n");

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

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

                        if (dist[i][j] == INF)

                                printf("INF ");

                else

                                printf("%3d ", dist[i][j]);

                        }

                        printf("\n");

        }

}

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];
// Update the distance matrix using intermediate vertices
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];
                }
        }
        }
        printSolution(dist);
}
int main() {
int graph[V][V] = {
        {0,     3,   INF,   7},
        {8,     0,     2,   INF},
        {5,   INF,   0,     1},
        {2,   INF, INF,   0}
};
```

```
    floydWarshall(graph);

    return 0;
}
```

## Output :




## Time Complexity:

Each of the three loops (k, i, j) runs from 0 to V-1.

So total steps ≈ $V \times V \times V = V^3$

Time Complexity = $O(V^3)$

## Space Complexity:

We store a $V \times V$ matrix (dist[V][V]).

Space Complexity = $O(V^2)$

## Real-Applications :

1. Google Maps / GPS Navigation

Used to find the shortest travel distance between all pairs of cities or locations.

Helps compute the minimum route between any two points on a map.

2. Airline Route Planning

Used to determine minimum connecting flight distances between airports.

Helps airlines optimize flight routes and fuel usage.

3. Social Network Analysis

Used to find the shortest link or connection between two people in a social graph.

Measures how closely users are connected (like degrees of separation).

4. Urban Transportation Systems

Used to design shortest bus, metro, or road paths in cities.

Improves time efficiency and reduces congestion.

5. Game Development (AI Pathfinding)

Used by non-player characters (NPCs) to find the shortest route to reach a target.

Improves movement and strategy in games.