

Name: Snehal Jayprakash Borji

UID: 2023510008

Course: F.Y.M.C.A.

Subject: DS

Practical 08

Aim: Implement All pair shortest path

Coding:

Prac8.cpp

```
#include <iostream>

#include <vector>

#include <climits>

using namespace std;

const int INF = INT_MAX;

void floydWarshall(vector<vector<int>>& graph, int vertices) {

    // Initialize the distance matrix

    vector<vector<int>> distance(vertices, vector<int>(vertices, 0));

    for (int i = 0; i < vertices; ++i) {
```

```

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

            distance[i][j] = graph[i][j];

        }

    }

    // Update the distance matrix

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

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

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

                if (distance[i][k] != INF && distance[k][j] != INF &&

                    distance[i][k] + distance[k][j] < distance[i][j]) {

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

                }

            }

        }

    }

    // Print the shortest distances

    cout << "All Pairs Shortest Paths:" << endl;

```

```

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

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

            if (distance[i][j] == INF) {

                cout << "INF ";

            } else {

                cout << distance[i][j] << " ";

            }

        }

        cout << endl;

    }

}

int main() {

    int vertices, edges;

    cout << "Enter the number of vertices: ";

    cin >> vertices;

    vector<vector<int>>> graph(vertices, vector<int>(vertices, INF));

```

```
// Input the graph edges and weights

cout << "Enter the number of edges: ";

cin >> edges;


cout << "Enter the edges and weights (u v weight):" << endl;

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

    int u, v, weight;

    cin >> u >> v >> weight;

    graph[u][v] = weight;

}


floydWarshall(graph, vertices);


return 0;

}
```

OUTPUT :

```
● mca@mca-HP-280-G3-SFF-Business-PC:~/snehal$ g++ prac8.cpp
● mca@mca-HP-280-G3-SFF-Business-PC:~/snehal$ ./a.out
Enter the number of vertices: 4
Enter the number of edges: 6
Enter the edges and weights (u v weight):
0 1 2
0 2 5
1 2 1
1 3 4
2 3 3
3 0 7
All Pairs Shortest Paths:
13 2 3 6
11 13 1 4
10 12 13 3
7 9 10 13
```