

Assignment No. 13

Name : Manish Namdev Barage

PRN : 2252007

Batch : T7

Topic : All Pairs Shortest Path problem using Floyd's algorithm

Code :

```
#include<bits/stdc++.h>
using namespace std;

void shortest_distance(vector<vector<int>>& matrix,
vector<vector<int>>& pred) {
    int n = matrix.size();

    for(int i = 0; i < n; i++) {
        for(int j = 0; j < n; j++) {
            if(matrix[i][j] == -1) {
                matrix[i][j] = INT_MAX;
            }
            pred[i][j] = -1; // Initialize predecessors
        }
    }

    for(int k = 0; k < n; k++) {
        for(int i = 0; i < n; i++) {
            for(int j = 0; j < n; j++) {
                if(matrix[i][k] != INT_MAX && matrix[k][j] != INT_MAX)
                {
                    if(matrix[i][j] > matrix[i][k] + matrix[k][j]) {
                        matrix[i][j] = matrix[i][k] + matrix[k][j];
                        pred[i][j] = k;
                    }
                }
            }
        }
    }

    // Print the matrix after each iteration
    cout << "Matrix after iteration " << k + 1 << ":" << endl;
    for(int i = 0; i < n; i++) {
        for(int j = 0; j < n; j++) {
            if(matrix[i][j] == INT_MAX) {
                cout << "INF ";
            } else {
                cout << matrix[i][j] << " ";
            }
        }
    }
}
```

```

        cout << endl;
    }
    cout << endl;
}

int main() {
    int n;
    cout << "Enter the number of vertices: ";
    cin >> n;
    vector<vector<int>> matrix(n, vector<int>(n));
    vector<vector<int>> pred(n, vector<int>(n));

    // cout << "Enter the weighted adjacency matrix (" << n << "x" <<
n << "):" << endl;
    for(int i = 0; i < n; i++) {
        for(int j = 0; j < n; j++) {
            cout<<"Enter the weight of edge from "<<i<<" to "<<j<<":
";

            cin >> matrix[i][j];
            if(matrix[i][j] == -1) {
                matrix[i][j] = INT_MAX;
            }
        }
    }

    shortest_distance(matrix, pred);

    cout << "Shortest Distance Matrix:" << endl;
    for(int i = 0; i < n; i++) {
        for(int j = 0; j < n; j++) {
            if(matrix[i][j] == INT_MAX) {
                cout << "INF ";
            } else {
                cout << matrix[i][j] << " ";
            }
        }
        cout << endl;
    }

    return 0;
}

```

Output:

```
Enter the weight of edge from 1 to 1: 0
Enter the weight of edge from 1 to 2: 6
Enter the weight of edge from 2 to 0: -1
Enter the weight of edge from 2 to 1: -1
Enter the weight of edge from 2 to 2: 0
Matrix after iteration 1:
0 1 43
1 0 6
INF INF 0

Matrix after iteration 2:
0 1 7
1 0 6
INF INF 0

Matrix after iteration 3:
0 1 7
1 0 6
INF INF 0

Shortest Distance Matrix:
0 1 7
1 0 6
INF INF 0
PS D:\Third Year\DAA\LAB\Assign 10>
```