Floyd-Warshall

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

#define V 4
#define INF INT_MAX

void printSolution(int dist[][V]){
    cout<<"The following matrix shows the shortest distances between every pair of vertices
    for (int i = 0; i < V; i++){
        for (int j = 0; j < V; j++){
            if (dist[i][j] == INF)
                cout<<"INF"<<"   ";
            else
                cout<<dist[i][j]<<"  ";
        }
        cout<<endl;
    }
}

void floydWarshall (int graph[][V]){
    int dist[V][V], i, j, k;

    for (i = 0; i < V; i++)
        for (j = 0; j < V; j++)
            dist[i][j] = graph[i][j];


    for (k = 0; k < V; k++){
        for (i = 0; i < V; i++){
            for (j = 0; j < V; j++){
                if (dist[i][k] != INF && dist[k][j] != INF && dist[i][k] + dist[k][j] < dist
                    dist[i][j] = dist[i][k] + dist[k][j];
            }
        }
    }

    printSolution(dist);
}

int main(){
    /* Let us create the following weighted graph
            10
```

```
    (0)------->(3)
        |    /|\
    5 |  |
        |    | 1
    \|/  |
    (1)------->(2)
            3    */
    int graph[V][V] = { {0, 5, INF, 10},
                        {INF, 0, 3, INF},
                        {INF, INF, 0, 1},
                        {INF, INF, INF, 0}
                    };

    floydWarshall(graph);
    return 0;
}
```