**#include <bits/stdc++.h>**

**using namespace std;**

**#define V 4**

**#define INF 99999**

**void printSolution(int dist[][V]);**

**void floydWarshall(int dist[][V])**

**{**

**int i, j, k;**

**for (k = 0; k < V; k++) {**

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

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

**if (dist[i][j] > (dist[i][k] + dist[k][j])**

**&& (dist[k][j] != INF**

**&& dist[i][k] != INF))**

**dist[i][j] = dist[i][k] + dist[k][j];**

**}**

**}**

**}**

**// Print the shortest distance matrix**

**printSolution(dist);**

**}**

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

**{**

**cout << "The following matrix shows the shortest "**

**"distances"**

**" between every pair of vertices \n";**

**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;**

**}**

**}**

**int main()**

**{**

**int graph[V][V] = { { 0, 5, INF, 10 },**

**{ INF, 0, 3, INF },**

**{ INF, INF, 0, 1 },**

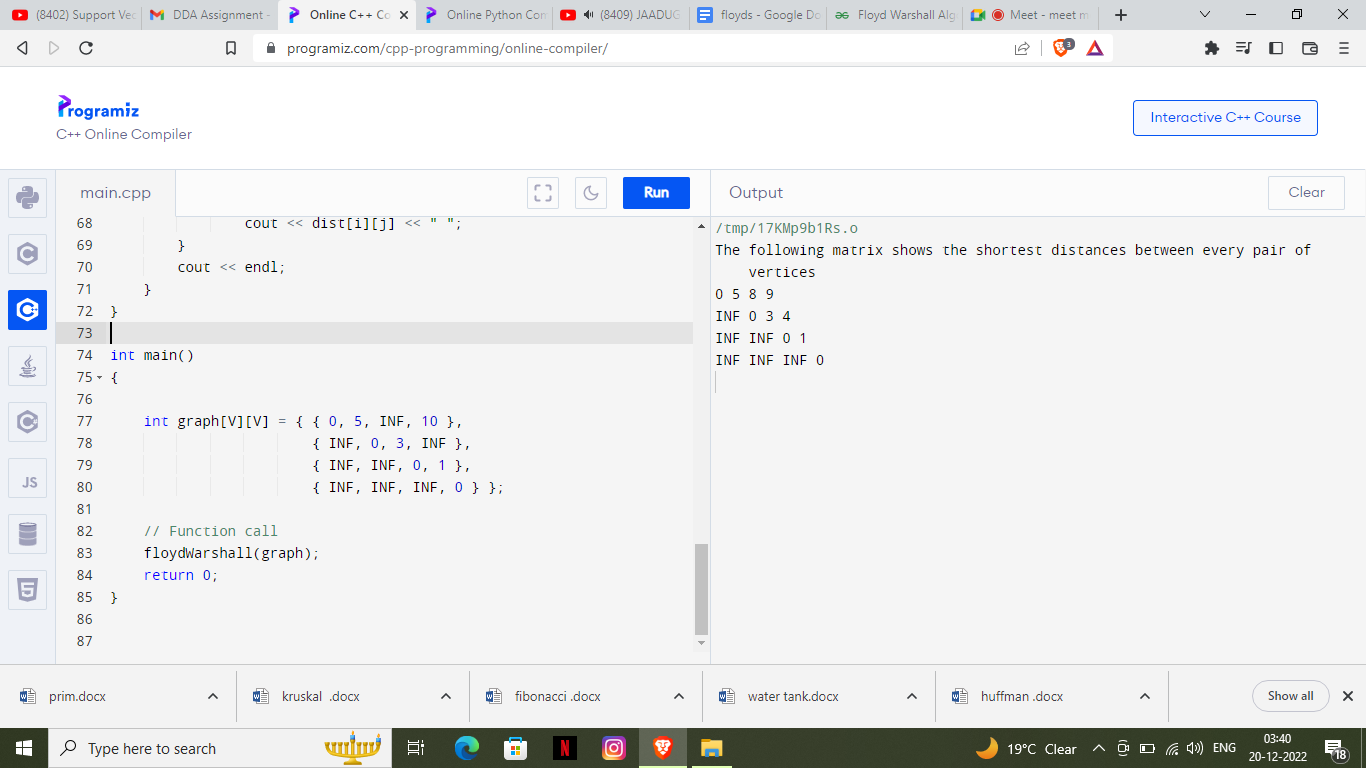
**{ INF, INF, INF, 0 } };**

**// Function call**

**floydWarshall(graph);**

**return 0;**

**}**

****