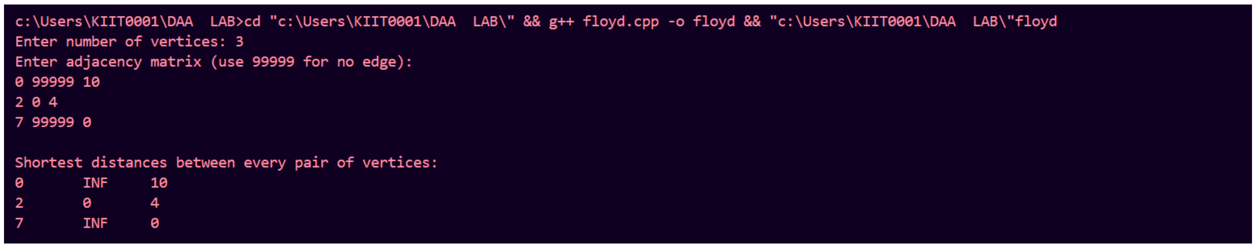
DAA LAB TASK 11

```cpp
#include <iostream>
#include <vector>
using namespace std;

#define INF 99999   // A large number to represent infinity

void floydWarshall(vector<vector<int>>& graph, int V) {
    // dist[][] will be the output matrix
    vector<vector<int>> dist = graph;

    // Floyd-Warshall Algorithm
    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] != INF && dist[k][j] != INF &&
                    dist[i][k] + dist[k][j] < dist[i][j])
                {
                    dist[i][j] = dist[i][k] + dist[k][j];
                }
            }
        }
    }

    // Print the result
    cout << "\nShortest 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\t";
            else
                cout << dist[i][j] << "\t";
        }
        cout << endl;
    }
}

int main() {
    int V;
    cout << "Enter number of vertices: ";
    cin >> V;

    vector<vector<int>> graph(V, vector<int>(V));

    cout << "Enter adjacency matrix (use " << INF << " for no edge):\n";
    for (int i = 0; i < V; i++) {
        for (int j = 0; j < V; j++) {
            cin >> graph[i][j];
        }
    }

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



```
c:\Users\KIIT0001\DAA  LAB>cd "c:\Users\KIIT0001\DAA  LAB\" && g++ floyd.cpp -o floyd && "c:\Users\KIIT0001\DAA  LAB\"floyd
Enter number of vertices: 3
Enter adjacency matrix (use 99999 for no edge):
0 99999 10
2 0 4
7 99999 0

Shortest distances between every pair of vertices:
0       INF     10
2       0       4
7       INF     0
```