```cpp
// AP21110010201
// Perumalla Dharan

#include <iostream>
#include <limits>
#include <vector>
using namespace std;
const int INF = numeric_limits<int>::max();
const int V = 4;

void printSolution(int dist[V][V]) {
    cout << "Distance Vector Routing table:\n";
    cout << "Source     Destination     Distance\n";
    for (int i = 0; i < V; i++) {
        for (int j = 0; j < V; j++) {
            if (dist[i][j] == INF)
                cout << i << "\t\t" << j << "\t\t\n";
            else
                cout << i << "\t\t" << j << "\t\t" << dist[i][j]
<< "\n";
        }
    }
    cout << "\n";
}

void distanceVectorRouting(int graph[V][V]) {
    int dist[V][V];
    for (int i = 0; i < V; i++) {
        for (int j = 0; j < V; j++) {
            dist[i][j] = graph[i][j];
        }
    }

    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];
                }
            }
        }
    }
    printSolution(dist);
}

int main() {
    int graph[V][V];

    cout << "Enter  matrix  (" << V << "x" << V << "):\n";
    for (int i = 0; i < V; i++) {
        for (int j = 0; j < V; j++) {
            cin >> graph[i][j];
            if (i == j) {
                graph[i][j] = 0;
            }
            if (graph[i][j] == 0) {
                graph[i][j] = INF;
            }
        }
    }
    distanceVectorRouting(graph);
    return 0;
}
```