Floyd’s Algorithm

**#include <stdio.h>**

**#define nV 4**

**#define INF 999**

**Void printMatrix(int matrix[][nV]);**

**Void floydWarshall(int graph[][nV]) {**

**Int matrix[nV][nV], i, j, k;**

**For (i = 0; i < nV; i++)**

**For (j = 0; j < nV; j++)**

**Matrix[i][j] = graph[i][j];**

**For (k = 0; k < nV; k++) {**

**For (i = 0; i < nV; i++) {**

**For (j = 0; j < nV; j++) {**

**If (matrix[i][k] + matrix[k][j] < matrix[i][j])**

**Matrix[i][j] = matrix[i][k] + matrix[k][j];**

**}**

**}**

**}**

**printMatrix(matrix);**

**}**

**Void printMatrix(int matrix[][nV]) {**

**For (int i = 0; i < nV; i++) {**

**For (int j = 0; j < nV; j++) {**

**If (matrix[i][j] == INF)**

**Printf(“%4s”, “INF”);**

**Else**

**Printf(“%4d”, matrix[i][j]);**

**}**

**Printf(“\n”);**

**}**

**}**

**Int main() {**

**Int graph[nV][nV] = {{0, 3, INF, 5},**

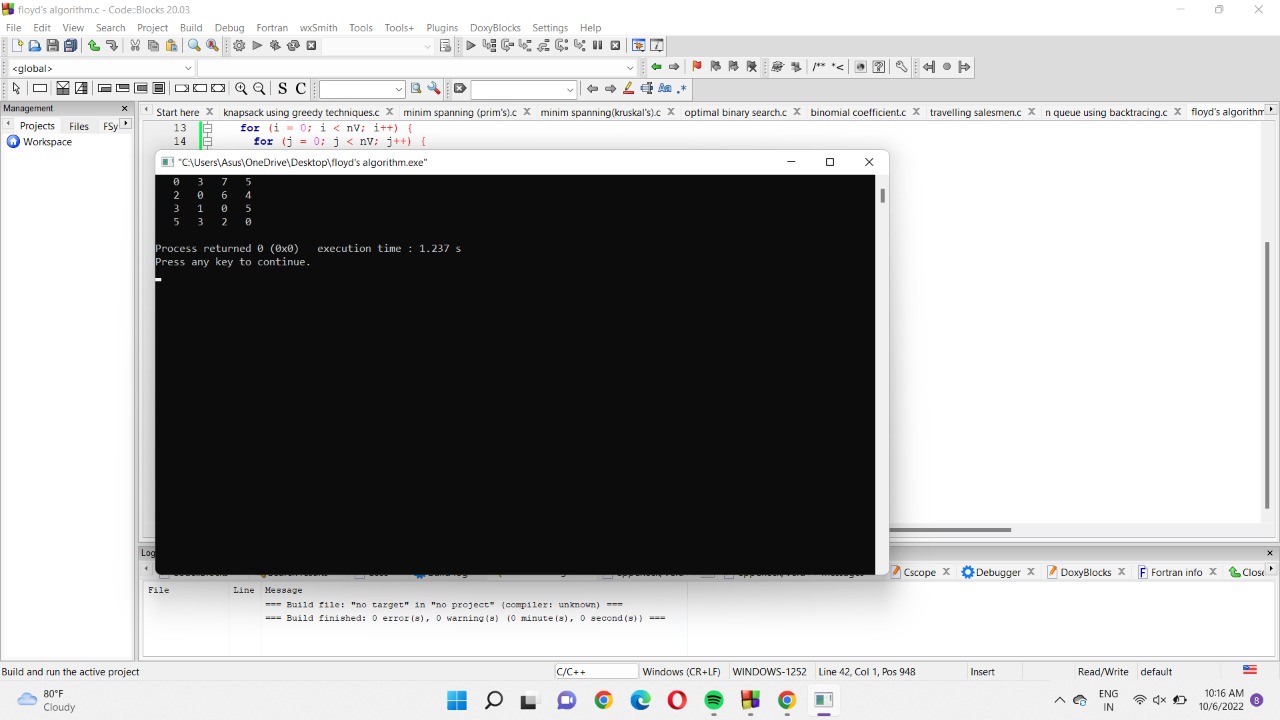
**{2, 0, INF, 4},**

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

**{INF, INF, 2, 0}};**

**floydWarshall(graph);**

**}**

****