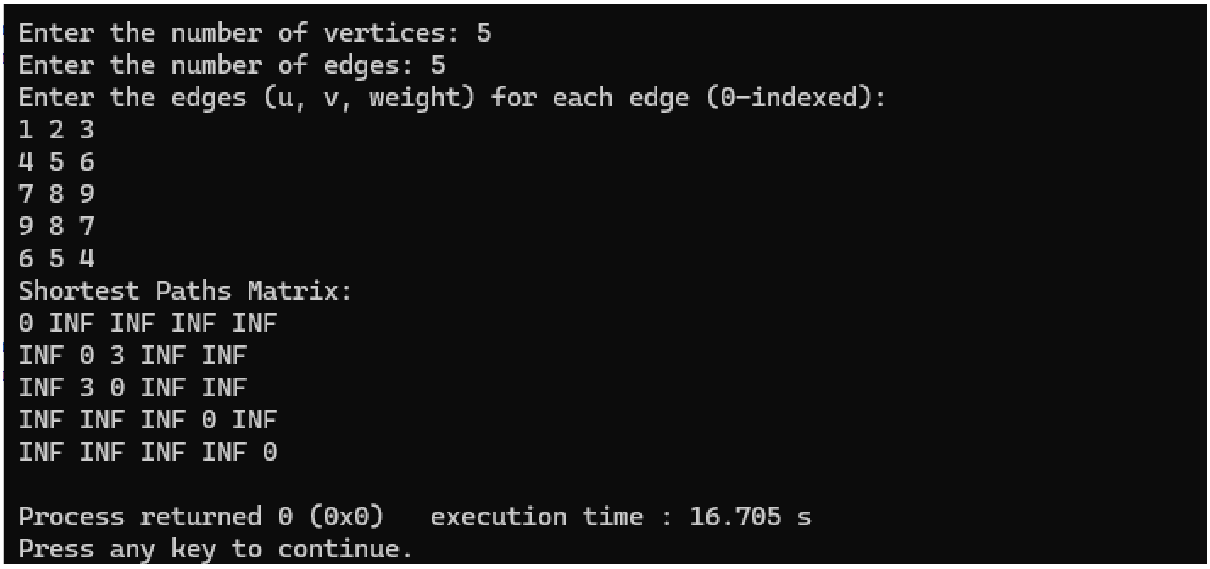
# FLOYDS

```
#include <stdio.h>
#include <limits.h>

#define MAX_VERTICES 10
#define INF INT_MAX

void floydWarshall(int graph[MAX_VERTICES][MAX_VERTICES], int V) {
    int dist[MAX_VERTICES][MAX_VERTICES];

    for (int i = 0; i < V; i++) {
        for (int j = 0; j < V; j++) {
            if (graph[i][j] == 0 && i != j) {
                dist[i][j] = INF;
            } else {
                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][j] > dist[i][k] + dist[k][j]) {
                    dist[i][j] = dist[i][k] + dist[k][j];
                }
            }
        }
    }
    printf("Shortest Paths Matrix:\n");
    for (int i = 0; i < V; i++) {
        for (int j = 0; j < V; j++) {
            if (dist[i][j] == INF) {
                printf("INF ");
            } else {
                printf("%d ", dist[i][j]);
            }
        }
        printf("\n");
    }
}

int main() {
    int V, E;

    printf("Enter the number of vertices: ");
    scanf("%d", &V);

    int graph[MAX_VERTICES][MAX_VERTICES] = {0};

    printf("Enter the number of edges: ");
    scanf("%d", &E);

    printf("Enter the edges (u, v, weight) for each edge (0-indexed):\n");
    for (int i = 0; i < E; i++) {
        int u, v, weight;
        scanf("%d %d %d", &u, &v, &weight);
        graph[u][v] = weight;
        graph[v][u] = weight;
    }

    floydWarshall(graph, V);

    return 0;
}
```

## OUTPUT:



```
Enter the number of vertices: 5
Enter the number of edges: 5
Enter the edges (u, v, weight) for each edge (0-indexed):
1 2 3
4 5 6
7 8 9
9 8 7
6 5 4
Shortest Paths Matrix:
0 INF INF INF INF
INF 0 3 INF INF
INF 3 0 INF INF
INF INF INF 0 INF
INF INF INF INF 0

Process returned 0 (0x0)   execution time : 16.705 s
Press any key to continue.
```