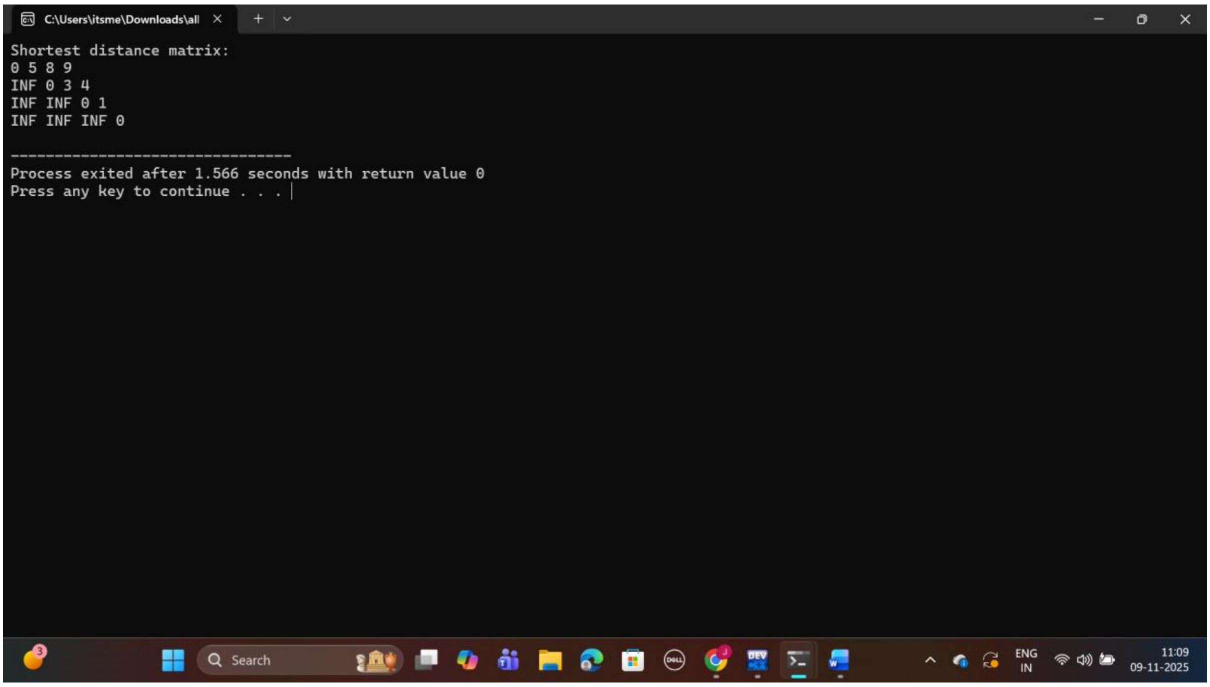
All-Pair Shortest Path (Floyd Warshall Algo.)

```
#include <bits/stdc++.h>
using namespace std;

const int INF = 1e9;

void floydWarshall(vector< vector<int> >& dist) {
    int n = dist.size();

    for (int k = 0; k < n; ++k) {
        for (int i = 0; i < n; ++i) {
            for (int j = 0; j < n; ++j) {
                if (dist[i][k] < INF && dist[k][j] < INF)
                    dist[i][j] = min(dist[i][j], dist[i][k] + dist[k][j]);
            }
        }
    }

    // Detect negative weight cycles
    for (int i = 0; i < n; ++i)
        if (dist[i][i] < 0)
            cout << "Negative weight cycle detected!\n";
}

int main() {
    int n = 4;
    vector< vector<int> > dist(4, vector<int>(4, INF));
    dist[0][0] = 0; dist[0][1] = 5;  dist[0][3] = 10;
    dist[1][1] = 0; dist[1][2] = 3;
    dist[2][2] = 0; dist[2][3] = 1;
    dist[3][3] = 0;

    floydWarshall(dist);

    cout << "Shortest distance matrix:\n";
    for (int i = 0; i < n; ++i) {
        for (int j = 0; j < n; ++j) {
            if (dist[i][j] == INF)
                cout << "INF ";
            else
                cout << dist[i][j] << " ";
        }
        cout << endl;
    }

    return 0;
}
```



```
Shortest distance matrix:
0 5 8 9
INF 0 3 4
INF INF 0 1
INF INF INF 0

--------------------------------
Process exited after 1.566 seconds with return value 0
Press any key to continue . . .
```