/\* TSP \*/

#include <iostream>

#include <vector>

#include <algorithm>

#include <limits>

using namespace std;

const int INF = numeric\_limits<int>::max();

int tsp(vector<vector<int>>& graph, int start, int mask, vector<vector<int>>& dp) {

int n = graph.size();

if (mask == (1 << n) - 1) {

return graph[start][0];

}

if (dp[start][mask] != -1) {

return dp[start][mask];

}

int ans = INF;

for (int i = 0; i < n; ++i) {

if ((mask & (1 << i)) == 0) {

int newMask = mask | (1 << i);

ans = min(ans, graph[start][i] + tsp(graph, i, newMask, dp));

}

}

return dp[start][mask] = ans;

}

int main() {

int n;

cout << "Enter the number of cities: ";

cin >> n;

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

cout << "Enter the cost matrix:" << endl;

for (int i = 0; i < n; ++i) {

for (int j = 0; j < n; ++j) {

cin >> graph[i][j];

}

}

vector<vector<int>> dp(n, vector<int>(1 << n, -1));

int minCost = tsp(graph, 0, 1, dp);

cout << "Minimum cost to visit all cities: " << minCost << endl;

return 0;

}

/\* OUTPUT

Enter the number of cities: 4

Enter the cost matrix:

0 10 15 20

10 0 35 25

15 35 0 30

20 25 30 0

Minimum cost to visit all cities: 80 \*/