#include <iostream>

#include <fstream>

#include <limits>

using namespace std;

int main() {

int n1, n2, i1, i2, j1, j2;

ifstream inputFile("input.txt");

if (!inputFile.is\_open()) {

cerr << "Error opening file!" << endl;

return 1;

}

inputFile >> n1 >> n2;

if (n1 > 20 || n2 > 20 || n1 <= 0 || n2 <= 0) {

cerr << "Invalid array dimensions!" << endl;

return 1;

}

int arr[20][20];

for (int i = 0; i < n1; ++i) {

for (int j = 0; j < n2; ++j) {

inputFile >> arr[i][j];

}

}

inputFile.close();

cout << "Enter i1, j1, i2, j2: ";

cin >> i1 >> j1 >> i2 >> j2;

if (i1 < 0 || i1 >= n1 || j1 < 0 || j1 >= n2 || i2 < 0 || i2 >= n1 || j2 < 0 || j2 >= n2) {

cerr << "Invalid start or end coordinates." << endl;

return 1;

}

long long dist[20][20];

int path[20][20][2];

int result\_arr[20][20];

bool visited[20][20] = { false }; // Массив для отметки посещенных клеток

for (int i = 0; i < n1; ++i) {

for (int j = 0; j < n2; ++j) {

dist[i][j] = numeric\_limits<long long>::max();

path[i][j][0] = -1;

path[i][j][1] = -1;

result\_arr[i][j] = arr[i][j];

}

}

dist[i1][j1] = 0;

visited[i1][j1] = true;

for (int count = 0; count < n1 \* n2; ++count) {

int best\_i = -1, best\_j = -1;

long long min\_dist = numeric\_limits<long long>::max();

for (int i = 0; i < n1; ++i) {

for (int j = 0; j < n2; ++j) {

if (arr[i][j] != 0 && dist[i][j] < min\_dist && !visited[i][j]) {

min\_dist = dist[i][j];

best\_i = i;

best\_j = j;

}

}

}

if (best\_i == -1) break; // Нет непосещенных клеток с доступными путями

visited[best\_i][best\_j] = true;

int dx[] = { 0, 0, 1, -1 };

int dy[] = { 1, -1, 0, 0 };

for (int k = 0; k < 4; ++k) {

int ni = best\_i + dx[k];

int nj = best\_j + dy[k];

if (ni >= 0 && ni < n1 && nj >= 0 && nj < n2 && arr[ni][nj] != 0) {

if (dist[best\_i][best\_j] + arr[ni][nj] < dist[ni][nj]) {

dist[ni][nj] = dist[best\_i][best\_j] + arr[ni][nj];

path[ni][nj][0] = best\_i;

path[ni][nj][1] = best\_j;

}

}

}

}

if (dist[i2][j2] == numeric\_limits<long long>::max()) {

cout << "No path found." << endl;

return 0;

}

cout << "Minimal sum: " << dist[i2][j2] << endl;

int cur\_i = i2, cur\_j = j2;

while (cur\_i != -1) {

result\_arr[cur\_i][cur\_j] \*= -1;

int prev\_i = path[cur\_i][cur\_j][0];

int prev\_j = path[cur\_i][cur\_j][1];

cur\_i = prev\_i;

cur\_j = prev\_j;

}

cout << "Result matrix:" << endl;

for (int i = 0; i < n1; ++i) {

for (int j = 0; j < n2; ++j) {

cout << result\_arr[i][j] << "\t";

}

cout << endl;

}

return 0;

}