

Homework: Problem Solving

This document defines the **homework assignments** for the ["Algorithms" course @ Software University](#). Please submit a single **zip / rar / 7z** archive holding the solutions (source code) of all below described problems.

Problem 1. Shortest Path in Matrix

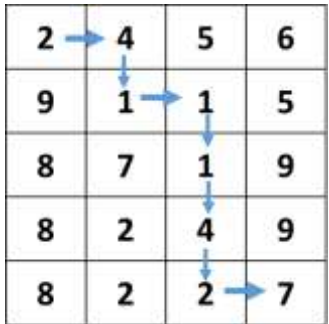
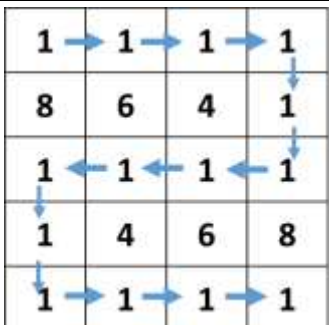
Write a program to find the **shortest path in a matrix of numbers** from the top-left corner to the bottom-right corner. The path consists of a sequence of cells, each sharing a common side with its next cell.

You will receive the number of **rows N** on the first line and the number of **columns M** on the second line. On each of the next N lines you'll receive the cells' values as a sequence of M **positive integers separated by a single space**.

Print the length of the path (sum of cell values) on the first line in format **"Length: {length}"**. On the second line, print the path in format **"Path: {cell1} {cell2} ..."**. You can test your solution in the Judge system [here](#).

Note: If multiple paths exist, print the one which moves through cells with lowest row and then column (traverse the matrix from top to bottom and from left to right).

Examples:

Input	Output	Path (Visualized)
5 4 2 4 5 6 9 1 1 5 8 7 1 9 8 2 4 9 8 2 2 7	Length: 22 Path: 2 4 1 1 1 4 2 7	
5 4 1 1 1 1 8 6 4 1 1 1 1 1 1 4 6 8 1 1 1 1	Length: 14 Path: 1 1 1 1 1 1 1 1 1 1 1 1 1 1	

5	Length: 13 Path: 1 1 4 1 1 1 1 1 1 1	<table><tr><td>1</td><td>→</td><td>1</td><td>1</td><td>1</td></tr><tr><td>8</td><td></td><td>↓</td><td>4</td><td>1</td></tr><tr><td>1</td><td>←</td><td>1</td><td>1</td><td>1</td></tr><tr><td>↓</td><td>1</td><td>4</td><td>6</td><td>8</td></tr><tr><td>↓</td><td>1</td><td>→</td><td>1</td><td>→</td><td>1</td><td>→</td><td>1</td></tr></table>	1	→	1	1	1	8		↓	4	1	1	←	1	1	1	↓	1	4	6	8	↓	1	→	1	→	1	→	1
1			→	1	1	1																								
8				↓	4	1																								
1			←	1	1	1																								
↓			1	4	6	8																								
↓			1	→	1	→	1	→	1																					
4																														
1 1 1 1																														
8 4 4 1																														
1 1 1 1																														
1 4 6 8																														
1 1 1 1																														

Hint: Build a graph and use **Dijkstra's algorithm**.