Grid

You are on an nxm grid where each square on the grid has a digit on it. From a given square that has digit k on it, a Move consists of jumping exactly k squares in one of the four cardinal directions. A move cannot go beyond the edges of the grid; it does not wrap. What is the minimum number of moves required to get from the top-left corner to the bottom-right corner?

Input

Each input will consist of a single test case. Note that your program may be run multiple times on different inputs. The first line of input contains two space-separated integers n and m ($1 \leq n, m \leq 500$), indicating the size of the grid. It is guaranteed that at least one of n and m is greater than 1. The next n lines will each consist of m digits, with no spaces, indicating the nxm grid. Each digit is between 0 and 9, inclusive. The top-left corner of the grid will be the square corresponding to the first character in the first line of the test case. The bottom-right corner of the grid will be the square corresponding to the last character in the last line of the test case.

Output

Output a single integer on a line by itself representing the minimum number of moves required to get from the top-left corner of the grid to the bottom-right. If it isn't possible, output -1.

Example

Input:
5 4
2120
1203
3113
1120
1110

Output:
6