Goal

It is well known, it is dangerous to leave alone to adventure without equipment. Before going to explore distant lands, your first test as an adventurer is thus to steal a sword in a trapped treasure room, since no old sage has kindly given you one.

The treasure room is represented by a grid whose entrance is in the upper left corner. The sword is in the lower right corner. The cells are either walls, empty or trapped cell: the floor collapses under your feet when you walk on a trapped box. Fortunately, you run fast enough to quickly move to the next cell of your path and do not fall with the floor, but once you've walked on such a cell, it becomes impassable. Consequently, when you are on a cell, you can move on one of the 4 adjacent cells as long as it is an empty cell or a trapped cell on which you have never passed.

Your goal is to go from the entrance of the hall to the sword, and then come back alive to the entrance. You can cross each trapped cell at most once during your round trip. You are also asked to minimize the number of trapped cells on your way.

Note: An algorithm of polynomial complexity is expected in the worst case.

Data

Input

Row *1*: an integer number **N** comprised between *3* and *20* representing the length and the width of the grid.
Rows *2* to **N**+*1*: the lines of the map represented by strings of **N** characters. The characters in the line are either # (wall) or . (empty cell), or ! (trapped cell). The cells of the entrance and the sword, located respectively at the top left and at the bottom right, will always be empty.

Output

An integer, indicating the minimum number of trapped cells to cross in order to take the sword and bring it back to the entrance. If it is possible to reach the sword but not come back alive then return -1. If it is not possible to reach the sword, return -2.

You can download sample input and output data files to work locally by clicking on the link at the bottom of the French version of the question

Téléchargez des fichiers d'exemple ainsi qu'un modèle de code pour travailler localement.