

Problem 2. Game map

Input file: .txt
Output file: output.txt
Time limit: 3 seconds (6 seconds for Java)
Memory limit: 256 mebibytes

Vasya still likes to play computer games, and he's still employed as a game tester. He's lost all hope of doing anything remotely interesting at work. To realize his creative potential, he is cooperating with his colleagues Petya and Kolya to work on a hobby project. Petya has told Vasya that there are a number of great indie games out there, and even a small team can do well in that business. He's not being entirely honest here — he hasn't told the guys about what percentage of those indie games gains any popularity or how many of those projects are completed at all.

The new game will be in the popular roguelike genre. The bearded Kolya keeps correcting Vasya: it's roguelite or roguelikelike. One of the most important components of a roguelike game is the randomly generated gameworld. The new adventure at each launch is the key to making "Losing is fun" motto work. The friends are working now on world map generation for their game.

The game takes place in caves. The world map is a rectangular field of cells. Each cell is either traversable, containing a level that must be passed by the player, or non-traversable and does not contain anything. The player begins the game at the top left corner cell, which must be traversable. Having finished a level, the player can move down or right to an adjacent cell, if that cell is traversable. While in the last row or column of the field, the player can leave the field. This ends the game, and the player wins.

For the purpose of balance, Petya has placed additional limitations on the map:

1. The player can reach each traversable cell from the starting cell though only one path.
2. The player can finish the game starting from any traversable cell, however, there can be several ways to do that.

Kolya is the chief programmer of the game and he will write the generation code. He has noticed that if the field is generated randomly, it almost never satisfies the criteria. For this reason, he's randomly generating only some of the field cells, leaving the rest in an indefinite state. Next he must run an algorithm that defines the contents of the remaining cells in such a manner that the resulting map satisfies all of the conditions. To increase the amount of content in the game, the algorithm must maximize the number of traversable cells.

The trouble is, Kolya is not a big expert on algorithmic programming and cannot implement the last step on his own. Help the friends make their dream come true!

Input

The first line of the input file contains two integers defining the size of the game field: H — the number of rows and W — the number of columns ($1 \leq H, W \leq 18$).

It is followed by the field contents as H lines with W symbols in each. Traversable cells are defined by the period symbol '.' (ASCII 46), non-traversable — by the letter 'X' (ASCII 88), and indefinite — by the question mark '?' (ASCII 63).

Output

If it is impossible to define the game field to a correct state, print the number -1 in the output file.

Otherwise, in the first line of the output file print an integer K — the number of traversable cells in the defined game field. This number must be as large as possible. Next, print the defined game field in the same format as that used in the input data — naturally, without the question marks.

If there are several optimal solutions, print any of them.

Examples

input.txt	output.txt
4 6 ??X??? .????? ?????.. ?X?X?.	13 .XXXXXX .X.X.. .X.XX.
3 3 ??X ?X? X??	-1