

Dangerous Parkour (parkour)

James wants to play his new videogame *Dangerous Parkour (DP)*. But after booting up the game, he realized that the game is incomplete. In fact, the developer forgot to add the levels to the game, and the character is only allowed to move right!



Figure 1: James playing his new game.

A level in *DP* is an $N \times M$ grid of cells, where cell $(0, 0)$ is the top-left cell and cell $(N - 1, M - 1)$ is the bottom-right cell. Each cell is either empty ('.') or contains a wall ('#').

James starts at cell $(0, 0)$ and must reach cell $(N - 1, M - 1)$ by making some (possibly 0) moves to the right: at any point, if James is at cell (i, j) and cell $(i, j + 1)$ is empty, he can move to cell $(i, j + 1)$. He may then start to fall down to cell $(i + 1, j + 1)$, $(i + 2, j + 1)$, $(i + 3, j + 1)$, and so on, until he either stands on top of a wall or reaches the bottom of the grid. James is not allowed to make a right move while falling down.

The fall damage is very high, and James can survive a fall of at most 1 cell. Can you create a level that allows James to win (that is, reach cell $(N - 1, M - 1)$ without dying), or tell him that it is impossible? The player needs to start in position $(0, 0)$, so position $(0, 0)$ must be empty and position $(1, 0)$ (if it exists) must be a wall.

📎 Among the attachments of this task you may find a template file `parkour.*` with a sample incomplete implementation.

Input

The input file consists of:

- a line containing integers N, M .

Output

The output file must contain -1 if it is not possible to design a winning level, or N lines, each containing M characters '.' or '#' describing a winning level. If there are multiple winning levels, you can output any of them.

Constraints

- $1 \leq N, M \leq 800$.

Scoring

Your program will be tested against several test cases grouped in subtasks. In order to obtain the score of a subtask, your program needs to correctly solve all of its test cases. If there are multiple output that let James win, you can output any of them.

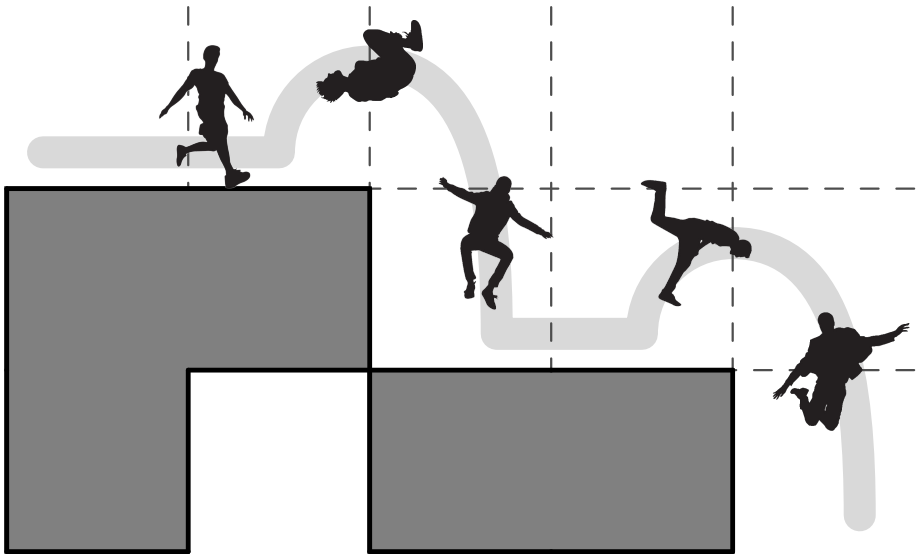
- Subtask 1 (0 points) Examples.
- Subtask 2 (30 points) $N, M \leq 5$.
- Subtask 3 (70 points) No additional limitations.

Examples

input	output
3 5 ##... #.#.#.
5 2	-1

Explanation

In the **first sample case** James wins following this path:



Note that this is not the only valid level, you can output any level that allows James to win.

In the **second sample case** it is impossible to create a level that allows James to win.