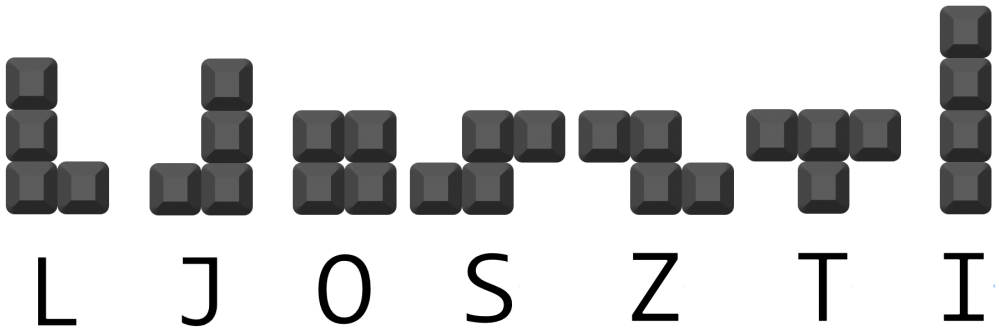
Elegant Tetris

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	512 megabytes

Tetris is a classical puzzle video game. In Tetris, players complete lines by descending differently shaped pieces (called tetrominoes) onto the playing field. The completed lines disappear and grant the player points.

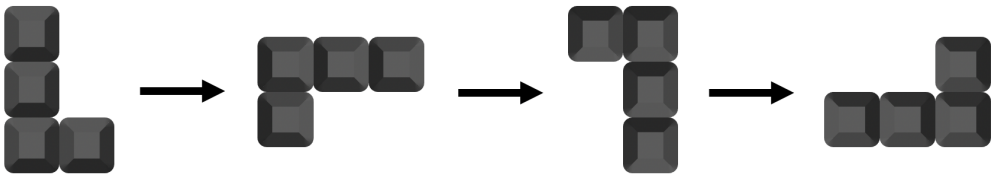
Elegant Tetris is a slightly different Tetris variant. In Elegant Tetris, you choose the shape, rotation and where to descend for each incoming piece.

There are 7 distinct shapes of tetrominoes, with each one represented by an uppercase letter, as in figure T-1.



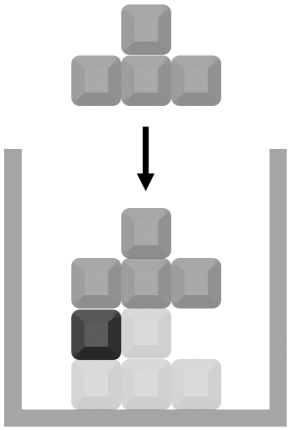
T-1

After deciding the shape of the piece, you may choose how many times to rotate the incoming piece. Each piece can be rotated at least 0 times and at most 3 times. Each rotation rotates the piece by 90° clockwise. For example, *O*-shaped pieces don't change after rotation, but *L*-shaped pieces change as in figure T-2.

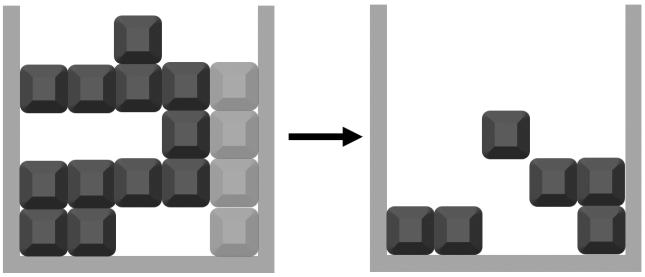


T-2

You may then decide where to descend the piece. The piece stops when it touches the bottom of the playing field or another piece. Note that in Elegant Tetris, you cannot move the piece horizontally during descending. For example, as shown in figure T-3, the *T*-shaped piece could be put under the 1x1 block in Tetris, but not in Elegant Tetris.



T-3



T-4

When the incoming block stops, all completed (fully filled) lines disappear, and the lines above fall, as figure T-4 shows. If after that there are still filled blocks with height **not less** than 20, game over.

Given an initial playing field with width w and some blocks already filled, Nikuniku wants to know how to go back to the initial playing field with no less than 1 and no more than 10000 moves without triggering game over.

Input

The first line contains two integers separated by a space w ($4 \leq w \leq 1000$) and n ($0 \leq n \leq 15$). w is the width of the playing field.

For the following n lines, each line contains a string with length w, denoting the lowest n lines of the playing field. The string consists of two symbols, '`#`' and '`.`'. '`#`' means the block is filled, and '`.`' means empty. No lines in the initial playing field are fully filled. All lines above height n are empty in the initial playing field.

It is guaranteed that the challenge is solvable. Note that it is **not** guaranteed that there is a movement sequence to go from the empty playing field to the initial playing field.

Output

On the first line, output an integer len ($1 \leq len \leq 10000$) – how many moves to make. Note that len does **not** have to be the minimal number of moves possible.

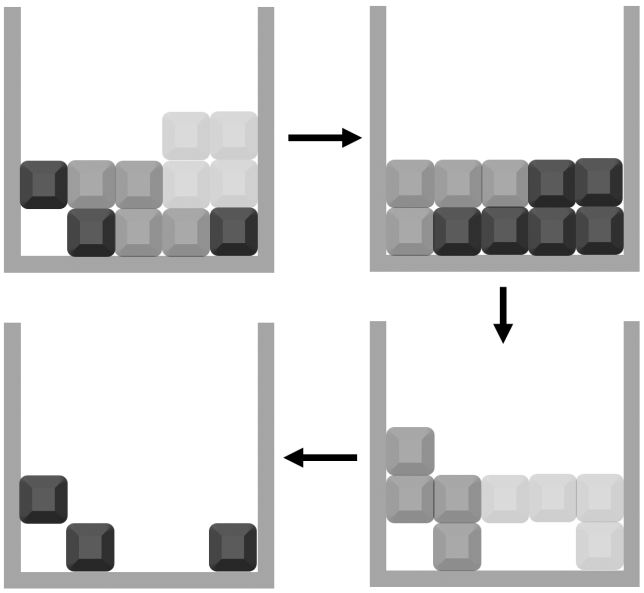
For the following len lines, output one uppercase letter ch ($ch \in \{$'`L`', '`J`', '`O`', '`S`', '`Z`', '`T`', '`I`'$\}$) and two integers a ($0 \leq a \leq 3$) and x ($1 \leq x \leq w$) separated by spaces – the shape, rotation count and the coordinate of the leftmost block, accordingly.

Examples

standard input	standard output
5 2 #.... .#..#	5 Z 0 2 O 0 4 L 1 1 S 1 1 J 3 3
5 4 #.... ###.. ####. #..#.	5 L 2 4 L 0 4 O 0 2 L 1 1 J 1 1

Note

The following figure shows the situation in the first example test case.



T-5