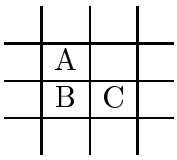
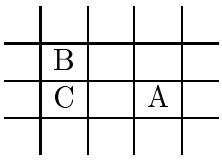
Problem B. Checkers

Input file:	checkers.in
Output file:	checkers.out
Time limit:	2 seconds
Memory limit:	256 megabytes

Peter likes to play with checkers on a chequered board. Recently he invented a new game: transform one beautiful picture composed of checkers into another beautiful picture. The following moves are allowed. Consider two following configurations of three selected cells each (let's denote the cells A, B and C):





.

If we select three fields arranged in any of the two ways shown above, you are allowed to do one of two possible moves. If cell A is occupied and cells B and C are free, it is allowed to remove checker from cell A and place two checkers into cells B and C. Vice versa, if cells B and C are occupied and cell A is free, it is allowed to remove checkers from cells B and C and place checker into cell A. Configurations can be taken in any place of the board, but can't be rotated or reflected. It is forbidden to make move if you try to remove checker from empty cell or if you try to place checker into occupied cell. Peter has such huge board that you can assume that it is infinite in any direction. Also you can assume that Peter has infinite number of checkers.

Peter decided to give riddle to his friend Vasya. He asked to transform one picture into another. Vasya tried different combinations of moves for a long time. Several times he got something simmilar to his goal, but not the same picture. Then Vasya decided that it's impossible to crack the riddle. But Peter said he has list of necessary moves.

Now Vasya asks you to write program which could help him solve Peter's puzzle.

Input

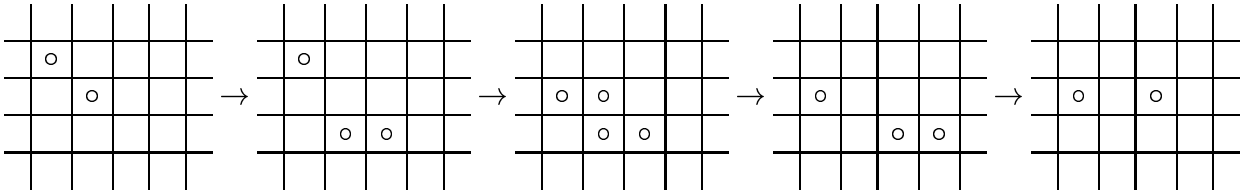
The first line of the input file contains N and M ($1 \le N, M \le 10$) — dimensions of the first picture. Following N lines describe the picture. Each line contain M symbols '`.`' or '`#`'. '`.`' means that cell is empty and '`#`' means that cell is occupied by checker. Description of the second picture in the same format follows immediately after first picture.

Output

Output sequence of moves that transforms first picture into second. Every line of ouput should contain three integer numbers — coordinates of cell A and number of configuration (1 or 2) for next move. Left bottom cell of each picture has coordinates $(0, 0)$ and right top cell has coordinates $(M-1, N-1)$. This cells are located in the same cell on the plane.

Example

checkers.in	checkers.out
2 2 #. .# 1 3 #.#	1 0 1 0 1 1 3 -1 2 2 0 1



One can prove that for any correct testcase there exists a sequence of less than 70 000 moves which solves this testcase.