

Move as in Checkers!

Consider a game played on a board by two players. The board consists of a $n \times m$ grid (n columns and m rows) composed of light and dark squares indexed by positive integers. Moves are made in turn by turn by both players who move their pawns (one player plays white, while the other plays black) forward using only the dark squares (the dark squares are those with an odd sum of indexes). To the white player "forward" means from the row with the smaller index to the row with larger index, while to the black player the meaning is reversed.

Two kinds of moves are allowed:

- Move one square diagonally forward to an unoccupied square.
- Move two steps forward in the same diagonal direction, jumping over the opponent's pawn on the intermediate square (capturing). It is allowed to capture only one pawn in one move. The captured pawn is then taken off the board.

Pawns may never be moved backwards or off the board, and if a pawn reaches the final line it just has to stay there.

Your task is give a longest possible sequence of moves that may have led from one given situation (called the *earlier* one) to another situation (called the *later* one).

Input data specification

The first line of the input contains two positive integers n ($3 \leq n \leq 50$) and m ($3 \leq m \leq 50$) - denoting the size of the board.

Then, the snapshot of the earlier situation is given, followed by a snapshot of the later situation. Each snapshot is a sequence of m lines of n characters, corresponding to the squares of a chessboard.

Character '.' - denotes an empty square, 'W' - a square with a white pawn, 'B' - a square with a black pawn. Assume that it is black's turn to move after the earlier position.

Output data specification

First, output the number k of moves which could have led from the earlier to the later position. In the next k lines, print the determined sequence of moves. Each move should be given in a separate line, using the format: `old_column old_row new_column` to describe the change of coordinates of a pawn.

Example 1

Input:

```
6 8
.W....
.....
.B....
W.....
...B..
```

..W...
B
B.

 .B....

 ...W..

Output:

11
 2 3 3
 2 1 4
 4 5 5
 4 3 6
 6 7 5
 3 6 4
 5 8 3
 1 4 2
 3 6 1
 6 5 4
 1 4 2

Scoring: This output would receive 11 points.

Comment: In the earlier situation white pawns occupy squares: (2, 1), (1, 4), (3, 6); while black occupy: (2, 3), (4, 5), (6, 7), (5, 8). After the first black move (2 3 3) all possible moves for white are: (2 1 1), (2 1 4) - capturing, (1 4 2), (3 6 2), (3 6 4).

Test data

All test data (prepared for testing programs during the series) are temporary and available below. The final assessment of solutions will be performed after the set series, using modified test data. Along with each data set the possible form of data changes has been described.

| t | $n \times m$ | W_b | B_b | W_e | B_e | Possible changes to the test data |
|---|--------------|-------|-------|-------|-------|---|
| 1 | 3x3 | 1 | 1 | 1 | 1 | W_e and B_e might be arbitrary changed. In the earlier situation, and likewise in the later situation, pawns might be arbitrary replaced. |
| 2 | 4x4 | 1 | 1 | 1 | 0 | W_e and B_e might be arbitrary changed. In the earlier situation, and likewise in the later situation, pawns might be arbitrary replaced. |
| 3 | 6x6 | 2 | 1 | 1 | 0 | W_e and B_e might be arbitrary changed. In the earlier situation, and likewise in the later situation, pawns might be arbitrary replaced. |
| 4 | 3x48 | 15 | 15 | 10 | 10 | In the earlier situation, and likewise in the later situation, pawns might be arbitrary replaced. The board might be resized by no more than two rows. |
| 5 | 48x3 | 7 | 7 | 5 | 5 | In the earlier situation, and likewise in the later situation, pawns might be arbitrary replaced. The board might be resized by no more than two columns. |

| | | | | | | |
|----|-------|----|----|----|----|---|
| 6 | 12x12 | 2 | 2 | 2 | 2 | In the earlier situation, and likewise in the later situation, pawns might be arbitrary replaced. |
| 7 | 12x12 | 3 | 3 | 2 | 2 | In the earlier situation, and likewise in the later situation, pawns might be arbitrary replaced. |
| 8 | 12x12 | 4 | 4 | 2 | 0 | In the earlier situation, and likewise in the later situation, pawns might be arbitrary replaced. |
| 9 | 14x14 | 6 | 6 | 3 | 4 | In the earlier situation, and likewise in the later situation, pawns might be arbitrary replaced. |
| 10 | 14x14 | 12 | 12 | 10 | 10 | In the earlier situation, and likewise in the later situation, pawns might be arbitrary replaced. |

- t - test number
- $n \times m$ - the size of the board
- W_b - the number of white pawns in the earlier situation
- B_b - the number of black pawns in the earlier situation
- W_e - the number of white pawns in the later situation
- B_e - the number of black pawns in the later situation

Scoring

The score of your program is the total of scores awarded for individual test cases. For each test case for which you find a solution in k moves you will receive k points.

The number of points given in the ranking is scaled so that it is equal to 10 for the contestant whose solution has the highest score, and is proportionally less for all solutions with lower scores.

Note

A [similar problem](#) appeared in the DASM Programming League 2004. Have a look at the description [of a possible solution](#) given by Adrian Kuegel.