

Peg Solitaire



Peg Solitaire is a 1-player game played with pegs placed on holes. A board consists of holes and pegs placed on some of the holes on the board. The objective of the game is to remove as many pegs as possible from the board by making a set of valid moves.

Each valid move requires the player to jump a peg orthogonally over an adjacent peg onto an empty hole two steps away in the same direction and remove the jumped peg.

Your task is to remove as many pegs as possible using a series of valid moves.

Input Format

The first line contains two integers M , N separated by a single space. M indicates the number of columns in the board and N indicates the number of rows.

* N lines follow each line having * M characters.

An empty hole is represented by "-" (ascii value: 45), a peg on the hole is represented by "." (ascii value: 46).

In addition to above, some parts of the board are inaccessible. They are represented by the character "x" (ascii value: 120)

Jump RIGHT

```
.. -> --.
```

Jump LEFT

```
-- -> ..
```

Jump UP

```
- .  
.-> -  
.-
```

Jump DOWN

```
. -  
.-> -  
- .
```

The above example shows the valid moves in each of the 4 directions.

The board is indexed at (0,0) in the top left and (M-1, N-1) at the bottom right. row increases from left to right and column increases from top to bottom.

Constraints

$5 \leq M, N \leq 10$

Output Format

Print P, P lines follow each line containing a valid move.

Each valid move is represented by two space separated integers which indicates the position of the peg that is to be moved followed by a MOVE where a move can be LEFT or RIGHT or UP or DOWN. You can stop your program early or print all the moves till no valid move can be made on the board.

Sample Input

```
7 7
XX...XX
XX...XX
.....
...-...
.....
XX...XX
XX...XX
```

Sample Output

```
2
1 3 RIGHT
2 1 DOWN
```

Explanation

For the sake of explanation, only 2 moves are printed. The first move takes the peg at (1,3) and makes a jump to its RIGHT. This will result in the following board

```
XX...XX
XX...XX
.....
.-....
.....
XX...XX
XX...XX
```

Now, the second move takes the peg at (2,1) and makes a jump DOWN.

```
XX...XX
XX-..XX
..-....
.-....
.....
XX...XX
XX...XX
```

More moves can be printed as long as a valid moves exists on the board. Lesser the pegs on the board, more is the score.

Scoring

There are 5 testcases in total. Each testcase carries a maximum score of 10.

score for each testcase = $(1 + \text{\#initialPegs} - \text{\#pegsLeft}) * 10 / \text{\#initialPegs}$