

## 3260 Link and Pop – the Block Game

Recently, Robert found a new game on the Internet that is the newest version of ‘Link and Pop’. The game rule is very simple. Initially, a board of size  $n \times m$  is filled with  $n \times m$  blocks. Each of these blocks has a symbol on it. All you need to do is to find a pair of blocks with the same symbol on them, which can be linked with a line that consists of at most three straight horizontal or vertical line segments.

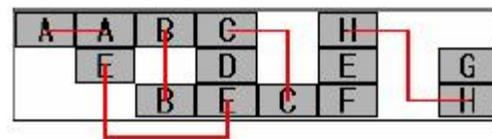


Fig.1

Note that the line segments cannot cross the other blocks on the board (see fig.1 for some examples of possible links, note that some blocks have been already removed from the board).

If you successfully find such a pair of blocks, the two blocks can be popped (that is, removed) together. After this, some of the blocks may be moved to new positions on the board following the rules described later. Then, you can start to find the next pair. The game continues until there are no block left on the board or you cannot find such a pair.

The blocks are moved according to the following rules. First, each block have a static moving attribute, which is one of ‘up’, ‘down’, ‘left’, ‘right’ and ‘stand still’. After a pair of block is removed, the blocks are checked one by one to see whether they can be moved towards the direction of its moving attribute. The blocks in the top row are checked first. Inside the same row, the blocks on the left are checked first. If the adjacent position at the direction of the block’s moving attribute is not occupied, the block will be moved to that position immediately. No block can be moved beyond the boundary of the game board. Of course, a block with attribute ‘stand still’ will always stay at its original position. After all the blocks are checked, which is called a turn of checking, another turn of checking is started. This continues until no more blocks can be moved to a new position following the moving rules. Note that inside each turn of checking, each of the blocks is checked and possibly moved only once. Blocks must not be checked and moved on its new position in one turn of checking.

Robert felt that the game was very interesting. However, after some time of playing, he found that when the size of the board is rather large, finding a pair of block becomes a very tough work. Further more, he often gets a ‘Game Over’ because of no more blocks can be popped. Robert felt that it is not his fault that not all the blocks are being popped. It is only that there is a great chance that the game cannot be finished if the blocks are placed randomly at first. However, it will be very time consuming to prove this by playing the game many times. So, Robert asks you to write a program for him that will simulate his behavior in the game and see if the game can be finished.

In order to make such a program possible, Robert summarizes his rules of selecting block pairs as follows. First, the pair of blocks that can be linked with one straight line segment must be found and popped first, because such kind of pairs are easy to find. Next, if such a pair does not exist, the pairs that can be linked by two straight line segments must be found and popped. Finally, if both of the two kinds of pairs do not exist, the pairs that can be linked by three straight line segments must be found and popped. If more than one pair that can be linked with the same number of straight line segments exists, the pair that contains a block, which is positioned at the most top row (or most left if two more blocks are positioned in the same row), will be selected first. If this rule still cannot break the tie (more than one pair may share one block that is positioned at the most top, left position), the other block in these pairs are compared according to the same rules. Fig.2 shows a trace of a mini game of ‘Link and Pop’ that follows the above rules.

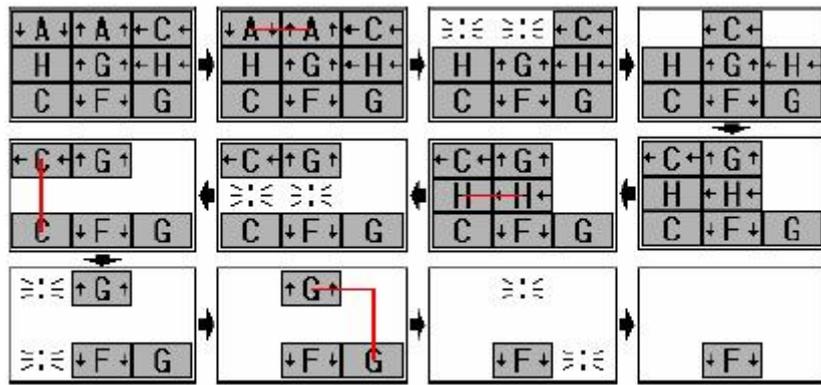


Fig.2

### Input

The input contains no more than 30 test cases. The first line of each test case contains 2 integers  $n, m$  ( $1 \leq n, m \leq 30$ ), which is the size of the board. After this line, there will be  $n$  more lines. Each of these lines contains  $m$  strings, separated by single spaces. Each of these strings represents one block in the initial configuration. Each string always consists of two capital letters. The first letter is the symbol of the block. The second letter is always one of the letters 'U', 'D', 'L', 'R' and 'S', which shows the block's moving attribute: up, down, left, right, and stand still respectively. There are no blank lines between test cases. The input ends with a line of two 0's: '0 0'.

### Output

For each test case, first output the test case number. After this line, you must output the final configuration of the board with  $n$  lines, each containing  $m$  characters. If there is a block on the position, output the symbol of the block. If there is no block on the position, output a period instead. Do not output blank lines between test cases.

### Sample Input

```
3 3
AD AU CL
HS GU HL
CS FD GS
1 2
BS BL
0 0
```

### Sample Output

```
Case 1
...
...
.F.
Case 2
..
```