Domino tiling

`tiling.c`, `tiling.C`, `tiling.java`, `tiling.p`

The aliens want to dominate the whole universe. So it comes as no surprise that their favorite game is played with *domino* tiles. A domino tile is a tile of size 1×2 with a digit 0-9 written on each half. Their game board is a rectangular array with a digit 0-9 written in each unit square. The task is to cover the whole board with the set of provided domino tiles. The tiles can be placed only if the two numbers on the tile equal the two numbers in the unit squares covered by the tile. The tiles can be placed as they are or may be rotated by 90°, 180° or 270°. No two tiles may overlap. You can always use only one piece of each provided tile.

Some tiles are already placed on the board. These must stay where they are.

You are to write a program that will play this game. If you do not succeed, our planet will be *domin*ated!

Input Specification

The input contains several descriptions of game settings. The first line of each description contains three numbers separated by one or more spaces. The first two numbers are the height M and the width N of the board. They satisfy $1 \leq M \leq 20$, $1 \leq N \leq 20$, at least one of them is even, and their product $2 \leq M \cdot N \leq 110$ (guess why). The last number is the number of available tiles K, $1 \leq K \leq M \cdot N/2$.

The next line contains K pairs of space-separated integers describing the pairs of numbers on the available tiles. Each two consecutive numbers on this line are separated by at least one space. In addition, no two tiles are identical, that is, they are different (and stay different even if one is rotated by 180°). The tiles already placed on the board are not among these K tiles.

The following M lines contain N space-separated entries each. For every $i, j, 0 \leq i < M, 0 \leq j < N$, the jth entry in the ith row describes the place in the ith row and jth column of the gameboard. The entry is either the capital letter "`X`" if there is an already-placed tile, or a number $A_{i,j}$ ($0 \leq A_{i,j} \leq 9$) written on the board.

Every description is followed by an empty line and the empty line after the last description is followed by a line containing three zeros.

Output Specification

For each game, find a way to place all tiles onto the board. If there are more solutions, output any of them. Provide the solution by a graphic representation as an $M \times N$ array with "`[`" and "`]`" (square brackets) standing for the left and right half of a horizontally placed domino, and lowercase letters "`n`" and "`u`" for the upper and lower half of a vertically placed domino. The squares covered by a tile in the input are still represented by "`X`". After the M rows, print one line containing the number of other *different* solutions that exist.

Do not output any spaces and start a new line for each row of the gameboard.

If there is no solution, write a single line with the word "`impossible`".

Print one empty line after the each gameboard result.

Sample Input

```
4 5 9
0 0  0 1  1 1  3 3  0 2  1 2    0 3   2 2   2 3
1 2 2 0 X
2 1 0 0 X
2 1 3 3 3
2 3 0 1 0

2 3 3
1 1 2 2 3 3
1 2 3
1 3 2

2 3 3
1 1 2 2 3 3
1 2 3
1 2 3

0 0 0
```

Output for Sample Input

```
[][]X
nn[]X
uun[]
[]u[]
3

impossible

nnn
uuu
0
```