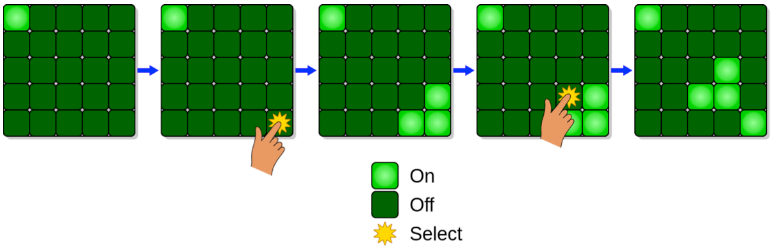
Technical challenge (“Lights out”)

There is a famous game called “Lights out” which starts with a board of lights (either on or off) and each time a light is switched the neighbouring lights are also switched. The goal of the game is to switch all the lights off. Below an image visualizing the gameplay.



(credits to Wikipedia)

The problem we describe below is a variation on this game. Instead of the plus-sign shape we introduce different “shapes”. And we also allow lights to go into more states then just on and off. This can be seen as turning the light from red, to green to blue back to red again. In this sense the goal would be to turn all lights red.

Problem

We start the puzzle with an initial board state where each cell has an initial value and a global “depth”. Each time we place a piece on the board and update the board state by incrementing each board cell which overlaps with a non-empty piece cell. If this value equals the “depth” of the board, then it is reset to 0. After all pieces have been placed, all cells should have the value 0.

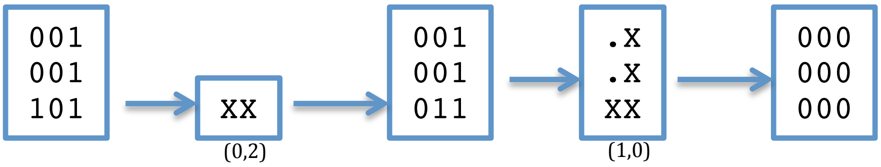
For example, we start with the following initial board state and pieces. We set the depth of the board to 2.

```
001                   .X
011         .X        .X
011         XX   XX   XX
board          pieces
```

We can start by placing the first piece in the left-bottom corner. This would give us the following board state:

```
001              001
011  →  .X   →   001
011     XX       101
       (0,1)
```

We switched two cells from 1 to 0 (1,1 and 1,2) and one cell from 0 to 1 (0,2). The other cell (0,1) was not touched. We place the remaining pieces in similar fashion on the board:



```
001          001    .X    000
001          001    .X    000
101    XX    011    XX    000
      (0,2)        (1,0)
```

Solved! All cells have the final state 0.

Some additional notes:

- Each piece must be placed on the board.
- Pieces cannot be placed outside the boundaries of the board.
- Pieces cannot be rotated, nor can the board.

Input

The input is a plain text file containing 3 lines:

- Line 1: "depth" of the puzzle. This will always be 2, 3 or 4.
- Line 2: initial board state. Each row is separated by a comma. Each digit represents the initial value for the cell.
- Line 3: individual pieces. Each piece is separated by a space. Each row within a piece is separated by a comma. '.' means no increment and 'X' means increment by 1.

```
2
001,011,011
.X,XX XX .X.X,XX
```

Output

The output should be printed to the standard out. Only one solution is required even though multiple might exist. The solution should be printed using the coordinate of each piece separated by a space. Each coordinate is formatted with "x,y" where the top-left corner of the board would be "0,0". The coordinate of the piece is always the top-left corner of the piece on the board, even if the top-left corner of the piece is empty. The order of the coordinates should be the same as the order of the pieces in the input file.

One possible solution for the above input is:

```
0,1 0,2 1,0
```