Game of Life 2020

Time limit: *2500 ms*
Memory limit: *256 MB*

To help verify the May 2020's Ponder This** challenge, May 2020 Ponder This, you need to write a simulator for a special version of Conway's game-of-life. In this game the board is cyclic (torus): the top of the board is connected to the bottom of the board and the left of the board is connected to the right of the board. We count the neighbors along common edges, so there are only 4 neighbors instead of the 8 neighbors found in the standard game-of-life.

Given the update rules for the empty cells and the live cells, and knowing the initial board, determine the final board state after a number of generations of the game.

Standard input

The input has the two update rules on the first line, separated by a semi-colon `;`. Each rule is given with exactly 5 bits of zeroes and ones. Empty cells should be updated using the first rule, and live cells should be updated using the second rule.

For empty cells, a `1` in the i-th bit of the rule sequence means that a live cell shall be born if exactly $i-1$ of its 4 neighbors are alive. Otherwise, the cell stays empty. For live cells, a `1` in i-th bit in the rule sequence means that a live cell shall stay live if exactly $i-1$ of its 4 neighbors are also alive. Otherwise, the live cell becomes empty.

The second line of the input contains two integers N and M, indicating that the board is of size $N \times N$ and the number of generations to simulate is M.

The next N lines each have N bits of zeroes and ones, giving the initial state of the board. A live cell is denoted by a `1`, and an empty cell is denoted by `0`.

Standard output

Output the final board state after M generations. The output should contain N rows with each row containing N bits of `0`s and `1`s.

Constraints and notes

- $3 \leq N \leq 25$
- $1 \leq M \leq 1\,000$

Input	Output	Explanation
00100;11000 3 100 000 010 000	000 010 000	The rules are that an empty cell becomes live if and only if it has 2 neighbors; and a live cell stays if and only if it has zero or one neighbors. So the board stays the same even after 100 generations.
01100;01100 4 1 0000 1000 1000 0000	1000 1101 1101 1000	Cells are born on all sides of the existing two cells. Since the board is cyclic, the left of the cells is at the rightmost of the board.
10000;01100 10 4 0000000000 0000000000 0000000000 0000000000 0000100000 0000000000 0000000000 0000000000 0000000000 0000000000	0000000000 0000000000 0010101000 0001010000 0010101000 0001010000 0010101000 0000000000 0000000000 0000000000	An empty board with this rule will oscillate between empty and full board. The single interference in the middle "explodes" outwards.
10000;01100 10 100 0000000000 0000000000 0000000000 0000000000 0000100000 0000000000 0000000000 0000000000 0000000000 0000000000	0010101000 0101010101 1000100010 0101010101 1010101011 0101010101 1000100010 0101010101 0010101000 0101110100	This is the same as the previous sample, but it runs for 100 generations.