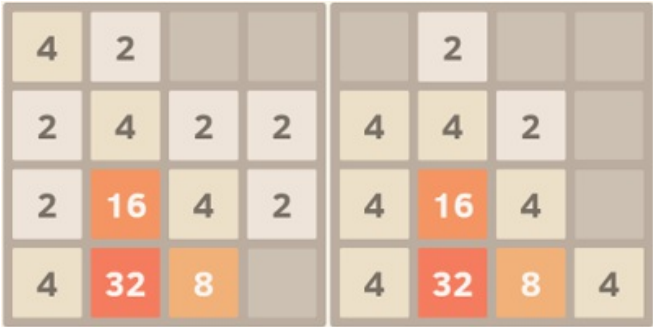


# TWENTY FORTY EIGHT

## CHALLENGE DESCRIPTION:

Have you ever tried to win in the 2048 game? In this challenge, your task is to implement part of the logic of this game.

The original 2048 is played on a 4×4 grid, with numbered tiles that slide when a player moves them using the four arrow keys. Tiles slide as far as possible in the chosen direction until they are stopped by either another tile or the edge of the grid. If two tiles of the same number collide while moving, they will merge into a tile with the total value of the two tiles that collided. The resulting tile cannot merge with another tile again in the same move (Description from Wikipedia [article](#)).



Picture from the [original game](#) (by Gabriele Cirulli)

In this challenge, you have to implement the logic for the sliding tiles in the chosen direction. Unlike the original game, a new tile cannot appear after the move and the grid size may vary from 4 to 10.

## INPUT SAMPLE:

The first argument is a path to a file. Each line of the input file contains one test case. Each test case consists of a direction (UP, DOWN, RIGHT, or LEFT), the length of a side in the square grid, and the initial configuration of the grid. All the data elements are separated by a semicolon. The configuration of the grid is a series of lines from top to bottom separated by a vertical bar. 0 indicates a blank tile.

For example:

```
RIGHT; 4; 4 0 2 0|0 0 0 8|4 0 2 4|2 4 2 2
UP; 4; 2 0 2 0|0 2 0 4|2 8 0 8|0 8 0 16
```

## OUTPUT SAMPLE:

For each test case, print to stdout the configuration of the grid received after slide of a tile in the chosen direction. Format is the same as in the input file. Lines are written from top to bottom and are separated by a vertical bar. Blank tiles are marked with 0.

For example:

```
0 0 4 2|0 0 0 8|0 4 2 4|0 2 4 4
4 2 2 4|0 16 0 8|0 0 0 16|0 0 0 0
```

## CONSTRAINTS:

- 1. The grid is square with the length of a side from 4 to 10.
- 2. The value of the tiles is equal to the power of number 2 and varies from 2<sup>1</sup> to 2<sup>10</sup>.