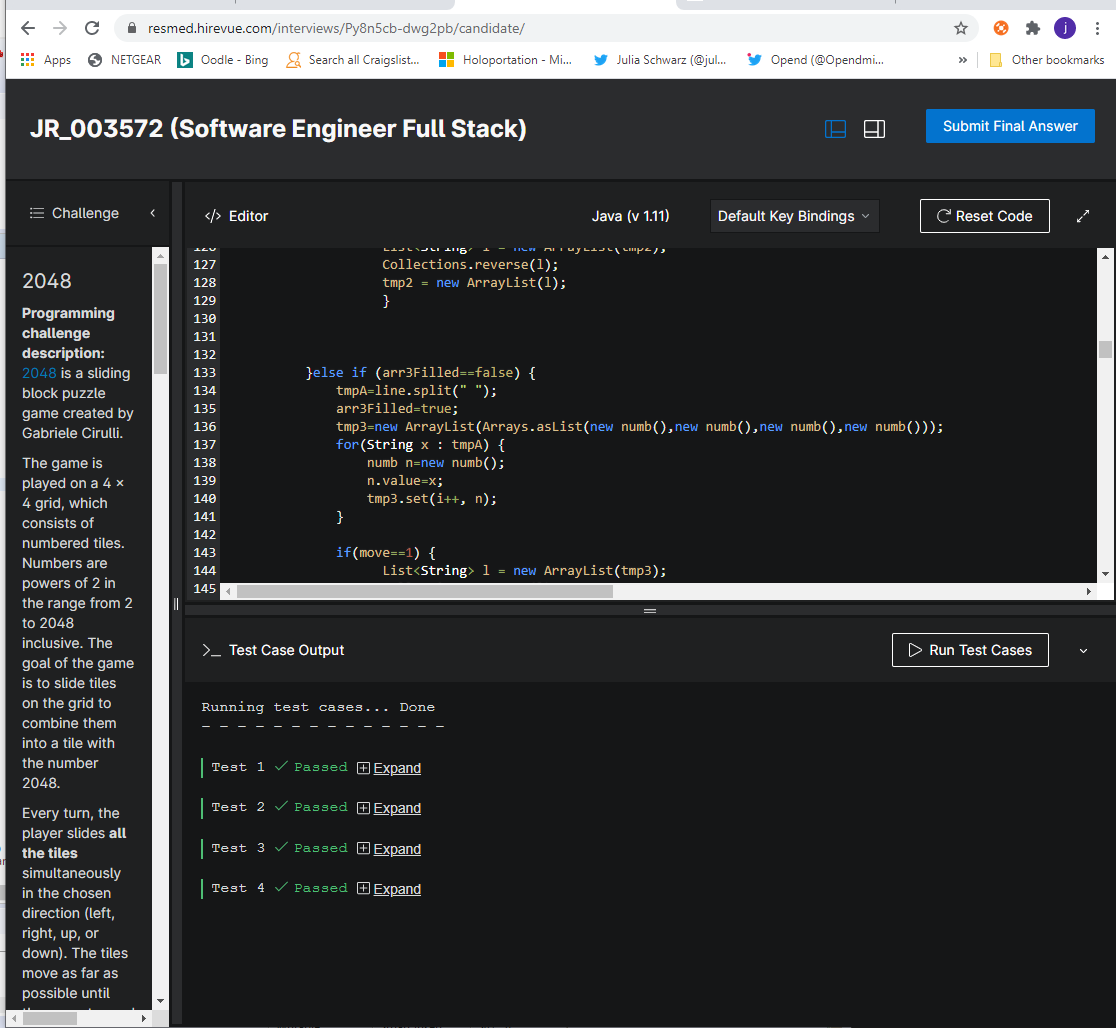
Result (4/4 sample tests passed)



Original specifications for code challenge:

2048

**Programming challenge description:**

[2048](https://2048game.com/) is a sliding block puzzle game created by Gabriele Cirulli.

The game is played on a 4 x 4 grid, which consists of numbered tiles. Numbers are powers of 2 in the range from 2 to 2048 inclusive. The goal of the game is to slide tiles on the grid to combine them into a tile with the number 2048.

Every turn, the player slides **all the tiles** simultaneously in the chosen direction (left, right, up, or down). The tiles move as far as possible until they are stopped by either another tile or the border of the grid. If two tiles of the same number collide while moving, they will merge into a single tile with the total value of the two tiles that collided. The resulting tile cannot merge with another tile again in the same move.

Note that if more than two tiles with the same numbers in one row are moved, the fusing process must be executed pairwise and starting from the grid's edge at the destination direction.

For example, if the player moves the row 2 2 2 2 to the right the result will be 0 0 4 4. And if the player moves the row 4 0 4 4 to the left the result will be 8 4 0 0.

**Input:**

The first line of input contains one of the string values left, right, up, or down to describes the direction of the move requested by the player.

The next four lines describe a valid state of the 2048 puzzle. Every line has four integers from the set \{0, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024\}{0,2,4,8,16,32,64,128,256,512,1024} and represents a corresponding row of the 4 x 4 grid. A cell with a zero is considered to be empty. For example:

right

0 4 0 4

32 32 32 16

8 8 8 8

2 0 0 4

**Output:**

Four lines with four integers representing the puzzle's state after the player's move is completed. Integers in the line must be space-delimited. Empty cells in the grid are described with zeroes.

In the real 2048 game, after each move, a new tile with a value of 2 or 4 will randomly appear in an empty cell. You can ignore this behavior in your solution and determine the state of the grid immediately before the new tile would appear. For example:

0 0 0 8

0 32 64 16

0 0 16 16

0 0 2 4

**Test 1**

**Test InputDownload Test 1 Input**

left 4 0 0 4

32 16 16 32

512 1024 1024

256 2 4 8 16

**Expected OutputDownload Test 1 Input**

8 0 0 0

32 32 32 0

512 2048 256 0

2 4 8 16

**Test 2**

**Test InputDownload Test 2 Input**

down 2 4 8 32

2 0 0 32

2 16 0 0

2 16 8 32

**Expected OutputDownload Test 2 Input**

0 0 0 0

0 0 0 0

4 4 0 32

4 32 16 64

**Test 3**

**Test InputDownload Test 3 Input**

up 2 4 8 16

32 64 128 256

512 1024 512 256

128 64 32 16

**Expected OutputDownload Test 3 Input**

2 4 8 16

32 64 128 512

512 1024 512

16 128 64 32 0

**Test 4**

**Test InputDownload Test 4 Input**

right 0 0 0 0

1024 0 1024 16

1024 2 1024 2

1024 512 0 512

**Expected OutputDownload Test 4 Input**

0 0 0 0

0 0 2048 16

1024 2 1024 2

0 0 1024 1024