⌃ **Easy**   ⓘ **Codewriting**   🔷 **1000**

Sudoku is a number-placement puzzle. The objective is to fill a `9 × 9` grid with numbers in such a way that each column, each row, and each of the nine `3 × 3` sub-grids that compose the grid *all* contain *all* of the numbers from `1` to `9` one time.

Implement an algorithm that will check whether the given `grid` of numbers represents a valid *Sudoku* puzzle according to the layout rules described above. Note that the puzzle represented by `grid` does not have to be solvable.

Example

- For

```
grid = [['.', '.', '.', '1', '4', '.', '.', '2', '.'],
        ['.', '.', '6', '.', '.', '.', '.', '.', '.'],
        ['.', '.', '.', '.', '.', '.', '.', '.', '.'],
        ['.', '.', '1', '.', '.', '.', '.', '.', '.'],
        ['.', '6', '7', '.', '.', '.', '.', '.', '9'],
        ['.', '.', '.', '.', '.', '.', '8', '1', '.'],
        ['.', '3', '.', '.', '.', '.', '.', '.', '6'],
        ['.', '.', '.', '.', '.', '7', '.', '.', '.'],
        ['.', '.', '.', '5', '.', '.', '.', '7', '.']]
```

  the output should be

  `sudoku2(grid) = true` ;

- For

```
grid = [['.', '.', '.', '.', '2', '.', '.', '9', '.'],
        ['.', '.', '.', '.', '6', '.', '.', '.', '.'],
        ['7', '1', '.', '.', '7', '5', '.', '.', '.'],
        ['.', '7', '.', '.', '.', '.', '.', '.', '.'],
        ['.', '.', '.', '.', '8', '3', '.', '.', '.'],
        ['.', '.', '8', '.', '.', '7', '.', '6', '.'],
        ['.', '.', '.', '.', '.', '2', '.', '.', '.'],
        ['.', '1', '.', '2', '.', '.', '.', '.', '.'],
        ['.', '2', '.', '.', '3', '.', '.', '.', '.']]
```

  the output should be

  `sudoku2(grid) = false` .

  The given `grid` is not correct because there are two `1` s in the second column. Each column, each row, and each `3 × 3` subgrid can only contain the numbers `1` through `9` one time.

Input/Output

- **[execution time limit] 20 seconds (scala)**

- **[input] array.array.char grid**

  A `9 × 9` array of characters, in which each character is either a digit from `'1'` to `'9'` or a period `'.'` .

- **[output] boolean**

  Return `true` if `grid` represents a valid *Sudoku* puzzle, otherwise return `false` .

[Scala] Syntax Tips

```scala
def helloWorld(name: String): String = {
    println("This prints to the console when you Run Tests")
    "Hello, " + name
}
```