Sudoku verifier

Sudoku is a game with few simple rules, where the goal is to place nine sets of positive digits (1...9) into the cells of a fixed grid structure (i.e. board).

The Sudoku board (or global grid) consists of a 3x3 arrangement of sub-grids, and each sub-grid is a 3x3 arrangement of cells. This yields a 9x9 arrangement of cells on the Sudoku board.

A valid Sudoku solution should conform to the following rules:

R1: A cell in a Sudoku game can only store positive digits, i.e. 1...9.

R2: All digits appear only once in a sub-grid, i.e. they cannot repeat.

R3: A digit can appear only once in the rows of the global grid.

R4: A digit can appear only once in the columns of the global grid.

• The program should read the candidate solution from a string variable, which should be exactly 81 characters long, i.e. first 9 are the first row, second 9 are the second row etc.

• Implement and document tests for method verify. Use boundary value analysis and the templates provided for Task 1.

• Familiarize yourself with Big List of naughty strings by Max Woolf: https://github.com/minimaxir/big-list-of-naughty-strings . Add a test case where you add a special Unicode character to the test string.

• Using parameterized tests is optional.

Program returns integer code based on the validity of the provided Sudoku solution:

Return 1 = solution contains characters which are not a number.

Return 0 = solution is valid Sudoku solution.

Return -1 = solution is violating R1

Return -2 = solution is violating R2

Return -3 = solution is violating R3

Return -4 = solution is violating R4

A correct Sudoku string:

417369825632158947958724316825437169791586432346912758289643571573291684164875293

An incorrect Sudoku string:

123456789912345678891234567789123456678912345567891234456789123345678912233456789122 34567891