



Un sudoku es un pasatiempo que consiste en rellenar una cuadrícula de 9x9 (81) casillas dividida en nueve regiones 3x3, tal como se explica a continuación. Los números del 1 a 9 se deben disponer de forma que no se repitan en ninguna fila, columna, o región del tablero. El sudoku se presenta inicialmente con algunas casillas rellenas, y el jugador debe completar los valores de las casillas vacías, que en nuestro caso particular representamos con ceros. **Para saber que un sudoku está correctamente resuelto basta con comprobar que en cada fila, columna y región están todos los números del 1 al 9.**

4	0	0		0	0	0		8	0	5
0	3	0		0	0	0		0	0	0
0	0	0		7	0	0		0	0	0

0	2	0		0	0	0		0	6	0
0	0	0		0	8	0		4	0	0
0	0	0		0	1	0		0	0	0

0	0	0		6	0	3		0	7	0
5	0	0		2	0	0		0	0	0
1	0	4		0	0	0		0	0	0

SUDOKU INCOMPLETO

4	1	7		3	6	9		8	2	5
6	3	2		1	5	8		9	4	7
9	5	8		7	2	4		3	1	6

8	2	5		4	3	7		1	6	9
7	9	1		5	8	6		4	3	2
3	4	6		9	1	2		7	5	8

2	8	9		6	4	3		5	7	1
5	7	3		2	9	1		6	8	4
1	6	4		8	7	5		2	9	3

SUDOKU RESUELTO

4	1	7		3	6	9		8	2	5
6	3	2		1	5	8		9	4	7
9	5	8		7	2	4		3	1	6

8	2	5		4	3	7		1	6	9
7	9	1		5	8	6		4	3	2
3	4	6		9	1	2		7	5	8

2	8	9		6	4	3		5	7	1
5	7	3		2	9	1		6	8	4
1	6	4		8	7	5		2	7	3

SUDOKU INCORRECTO

En los ejemplos mostrados, el sudoku de la izquierda está incompleto, el sudoku de en medio está resuelto correctamente, y el de la derecha es incorrecto porque en la fila 9 no está el número 9, o en la columna 8 no está el 9, o en la región novena (numerando de arriba abajo y de izquierda a derecha los cuadros), no está el 9. Nótese que las regiones se enumeran de izquierda a derecha y de arriba abajo.

Como parte de este ejercicio se proponen las siguientes tareas de complejidad creciente:

TAREA 1: Dado un sudoku representado por una matriz de enteros 9x9, escribir los siguientes métodos.

```
/**
 * leer_sudoku: Lee un sudoku introducido por el usuario vía teclado y lo
 * almacena en una matriz de enteros 9x9. La lectura de los datos será de
 * izquierda derecha y de arriba abajo.
 * @return Devuelve matriz de enteros 9x9 que representa el sudoku
 */
public static int[][] leer_sudoku ()

/**
 * escribir_sudoku: escribe un sudoku por pantalla en el formato adecuado
 * @param s: matriz de enteros 9x9 que representa el sudoku
 */
public static void escribir_sudoku (int[][] s)
```

TAREA 2: Dado un sudoku, escribir los siguientes métodos.

```

/**
 * esta_completo: Comprueba que un sudoku no tiene ceros.
 * @param s: matriz de enteros 9x9 que representa el sudoku
 * @return Devuelve true si no tiene ningún cero, false en caso contrario
 */
public static boolean esta_completo (int [][] s)

/**
 * fila_correcta: Comprueba que en la fila están los valores del 1 al 9.
 * @param s: matriz de enteros 9x9 que representa el sudoku
 * @param fila: índice de la fila a comprobar (de 0 a 8, de arriba a abajo)
 * @return Devuelve true si están del 1 al 9, y false en caso contrario */
public static boolean fila_correcta(int [][]s, int fila)

/**
 * region_correcta: Comprueba que en la región están los valores del 1 al 9
 * @param s: matriz de enteros 9x9 que representa el sudoku
 * @param region: índice de la región 3x3 a comprobar (de 0 a 8)
 * Las regiones se numeran de izq a der. y de arriba a abajo.
 *
 *      | |
 *      0 | 1 | 2
 *      | |
 * ---|---|---
 *      | |
 *      3 | 4 | 5
 *      | |
 * ---|---|---
 *      | |
 *      6 | 7 | 8
 *      | |
 *
 * @return Devuelve true si están del 1 al 9, y false en caso contrario */
public static boolean region_correcta(int [][]s, int region)

/**
 * es_correcto: Comprueba que un sudoku está correctamente resuelto.
 * @param s: matriz de enteros 9x9 que representa el sudoku
 * @return Devuelve true si está bien resuelto, y false en caso contrario
 */
public static boolean es_correcto(int [][] s)

```

NOTA: Podéis asumir que el método `columna_correcta`, equivalente a fila correcta, está implementado y puede ser reutilizado para programar el método `es_correcto`.

TAREA 3: Dado el tipo de datos Sudoku representado por la clase Sudoku:

```

public class Sudoku {
    final int FILAS=9;
    final int COLUMNAS=9;
    int [][] numeros;
    boolean completo;
    boolean correcto;
}

```



}

Escribir el método Java leerSudoku que lee de teclado una secuencia de 81 enteros separados por espacios en blanco o fines de línea y devuelve un dato de tipo Sudoku. El método almacena los enteros que representan el sudoku en el campo `numeros`, si está completo (no hay ningún cero) asigna **true** al campo `completo`, y en caso de que esté completo si es correcto asigna **true** al campo `correcto`. Puedes utilizar los métodos implementados en la primera Tarea.

```
/**
 * leeSudoku: Devuelve un Sudoku leído de teclado
 * @return Sudoku con los valores numéricos en el campo número,
 *         indicando si está completo, y en caso de que lo esté si
 *         está correctamente relleno
 */
public static Sudoku leerSudoku ()
```