deep se
dependable evolvable pervasive software engineering group

Lab session: C functions

Danilo Ardagna

Politecnico di Milano

danilo.ardagna@polimi.it

POLITECNICO DI MILANO

The problem - Sudoku

- In the Sudoku game, a 9x9 grid must be filled with numbers from 1 to 9
- The grid is split in nine rows, nine columns and nine 3x3 subregions; all rows, columns and subregions must contain exactly all numbers from 1 to 9

Example:

```
1 2 3 | 4 5 6 | 7 8 9
4 5 6 | 7 8 9 | 1 2 3
7 8 9 | 1 2 3 | 4 5 6
-------------------------
2 3 4 | 5 6 7 | 8 9 1
5 6 7 | 8 9 1 | 2 3 4
8 9 1 | 2 3 4 | 5 6 7
-------------------------
3 4 5 | 6 7 8 | 9 1 2
6 7 8 | 9 1 2 | 3 4 5
9 1 2 | 3 4 5 | 6 7 8
```

The problem - Sudoku

- **Goal:** write a program that checks whether a given Sudoku matrix is correct

- A search function

```
int search_key (const unsigned v[], unsigned n_elements,
                unsigned key);
```

that returns 1 if `key` is contained in the array `v`, 0 otherwise, is given

- To generate a Sudoku matrix, the given function

```
void generate_sudoku(unsigned sudoku[][SIZE]);
```

can be also used

The problem - Sudoku

You have to implement:

1) a function

```
int basic_search (const unsigned v[], unsigned n_elements);
```

that returns 1 if all numbers from 1 to 9 are contained in v, 0 otherwise

2) three functions

```
int check_rows (const unsigned sudoku[][SIZE]);
int check_cols (const unsigned sudoku[][SIZE]);
int check_regions (const unsigned sudoku[][SIZE]);
```

that return 1 if all the rows, columns and subregions (respectively) of the Sudoku matrix comply with the rules, 0 otherwise

The problem - Sudoku

3) a function

```
int check_sudoku(const unsigned sudoku[][SIZE]);
```

that returns:

- 1 if the given Sudoku matrix complies to all Sudoku rules
- -1 if a row violates the game rules
- -2 if a column violates the game rules
- -3 if a region violates the game rules

Credits

- Lewis' algorithm to generate a Sudoku matrix

https://en.wikipedia.org/wiki/Sudoku_solving_algorithms

Yoda's lesson