RESIT ASSIGNMENT

Issue Date : 29.01.2024 - Monday
Due Date : 02.02.2024 - Friday (23:00)
Advisor : R.A. Görkem AKYILDIZ
Programming Language : Python 3.6.8

1 Introduction

Sudoku is a logic-based, combinatorial number-placement puzzle. In classic Sudoku, the objective is to fill a 9 × 9 grid with digits so that each column, each row, and each of the nine 3 × 3 sub grids that compose the grid (also called "boxes", "blocks", or "regions") contains all of the digits from 1 to 9. The puzzle setter provides a partially completed grid, which for a well-posed puzzle has a single solution.[1]

Aim of this project is solving the Sudoku puzzles that probably require backtracking while solving it.

2 Definition of Input

Initial status of the Sudoku map will be given as a text file. Lines correspond to rows and each digit at the row corresponds to a cell. It can be interpreted as how it seems visually. Each cell is separated with space from each other for the sake of better visualization. Empty cells are represented with number zero instead of white space to make input file more understandable. You must take the name of the input file as the first command line argument of your program. A sample Sudoku puzzle and its text representation can be found below:

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

Figure 1: Puzzle

```
1  0 4 0 0 0 0 1 7 9
2  0 0 2 0 0 8 0 5 4
3  0 0 6 0 0 5 0 0 8
4  0 8 0 0 7 0 9 1 0
5  0 5 0 0 9 0 0 3 0
6  0 1 9 0 6 0 0 4 0
7  3 0 0 4 0 0 7 0 0
8  5 7 0 1 0 0 2 0 0
9  9 2 8 0 0 0 0 6 0
```

Figure 2: Text file for that puzzle

3 Definition of Output

Format of the output is the same as the input, you must printout completely solved version of the given sudoku puzzle. **No partial points will be given.** You must take the name of the output file as the second command line argument of your program.

4 Restrictions

- Your code must be able to execute on our department's developer server (dev.cs.hacettepe.edu.tr).
- You must obey given submit hierarchy and get score (1 point) from the submit system.
- **You must benefit from loops, functions, and recursion.**
- Your code must be clean, do not forget that main method is just a driver method that means it is just for making your code fragments run, not for using them as a main container, create functions in necessary situations but use them as required.
- You must use comments for this project and you must give brief information about the challenging parts of your code. Do not over comment as it is against clean code approach. Design your comments so that they make your code fully understandable and not excessive for others.

- You can benefit from Internet sources for inspiration but do not use any code that does not belong to you.
- You can discuss high-level (design) problems with your friends but do not share any code or implementation with anybody.
- Do not miss the submission deadline.
- Source code readability is a great of importance. Thus, write READABLE SOURCE CODE, comments, and clear MAIN function. This expectation will be graded as "clean code".
- Use UNDERSTANDABLE names to your variables, classes, and functions regardless of the length. The names of classes, attributes and methods should obey Python naming convention. This expectation will be graded as "coding standards".
- You can ask your questions through course's Piazza group, and you are supposed to be aware of everything discussed in the Piazza group. General discussion of the problem is allowed, but **DO NOT SHARE** answers, algorithms, source codes and reports.
- All assignments must be original, individual work. Duplicate or very similar assignments are both going to be considered as cheating.
- Submit system for this homework will be opened a few days before deadline, so please be patient.

5 Execution and Test

Your code must be executed under **Python 3.6.8** at **dev.cs.hacettepe.edu.tr**. If your code does not run at developer server during the testing stage, then you will be graded as 0 for code part even if it works on your own machine. Sample run command is as follows:

- python3 sudoku_backtrack.py input.txt output.txt

6 Submit Format

File hierarchy must be zipped before submitted (Not .rar, only not compressed .zip files because the system just supports .zip files).

```
- b<StudentID>.zip
    - sudoku_backtrack.py
```

7 Grading

Task	Point
Correct Output	80
Clean Code & Comment	20*
Total	100

*** The score of the part of clean code & comment will be multiplied by your overall score (excluding clean code & comment part) divided by the maximum score that can be taken from these parts. Say that you got 60 from all parts excluding clean code & comment part and 10 from clean code & comment part, your score for clean code & comment part is going to be 10*(60/80) which is 7.5 and your overall score will be 60+7.5=67.5.**
Note that you must score one at the submit system, otherwise 20% of your grade will be deducted, moreover, you must implement a main function otherwise 10% of your grade will be deducted! There may also be other point deductions if you do not obey the given rules, such as if you do not use functions, loops and/or recursion as necessary.

References

[1] Sudoku - wikipedia. `https://en.wikipedia.org/wiki/Sudoku` (Last access: 07.11.2023).