Ewen Quimerc'h project

Here is the docker image of my project: https://hub.docker.com/repository/docker/feust/sudoku-solver

What is it?

A simple sudoku solver. You can find the source code here

It is shipped in a **Docker image**, so you'll have to run a container (don't worry, the command is juste below). It **mounts** a folder from your computer to the corresponding container's folder containing the sudokus. I could have done it other ways (using just a string for example) but I did it for the challenge and explore a bit Docker's functionalities.

How to use it

You should:

- put all the sudoku you want to solve in a folder `/data`
- open a terminal
- go to the `/data` parent folder (so you can see the /data folder by typing `ls`)
- run the following command : `docker run --rm -v "$(pwd)/data:/mnt/data" feust/sudoku-solver:2.1 mon_sudoku.txt`

If you have more skills and you want to use it a more convenient way, you can make aliases (BEWARE it depends on your computer and config):

```
echo "alias solve="docker run --rm -v "$(pwd)/data:/mnt/data" feust/sudoku-solver:2.1"" >> ~/.bashrc
bash
solve  <name_of_your_sudoku.txt> <name_of-sudoku2.txt>
```

The results are displayed in your terminal AND in some files created on your /data folder 😃

Handled cases

I tried to handle as many cases I could. You can solve several sudokus in a row, or even all the sudokus in your folder by typing `$(ls)` instead of the names

Format

The sudokus should be saved in plain text files and look like this :

```
...546..9
.2......7
..39....4
9.5....7.
```

```
7......2.
....93...
.56..8...
.1..39...
......8.6
```

From command line instead of files

Just type something like this:

```
echo "...546..9
.2......7
..39....4
9.5....7.
7......2.
....93...
.56..8...
.1..39...
......8.6" > data/test && solve test
```

Dockerfile

```
FROM python:3.7

# Creating a special directory
RUN mkdir -p /solver

# Copying only the files we need
COPY main.py parser.py solver.py requirements.txt /solver/

# Working in solver
WORKDIR /solver

# Instaling the librairies (pip comes with the python image)
RUN pip install -r requirements.txt

# Solving ! Need arguments that will be stored in /mnt/data
ENTRYPOINT [ "python3", "main.py"]
```

Examples

For one file:

```
lanfeust@troy ~/dev/tests docker run --rm -v "$(pwd)/data:/mnt/data" feust/sudoku-solver:2.1 sudoku2.txt

=== Reading file /mnt/data/sudoku2.txt ===
[[1 7 8 5 4 6 2 3 9]
 [4 2 9 3 8 1 5 6 7]
 [5 6 3 9 2 7 1 8 4]
 [9 3 5 2 1 4 6 7 8]
 [7 4 1 8 6 5 9 2 3]
 [6 8 2 7 9 3 4 1 5]
 [2 5 6 4 7 8 3 9 1]
 [8 1 4 6 3 9 7 5 2]
 [3 9 7 1 5 2 8 4 6]]

Sudoku solving speed:
Parsing... 0.0001 seconds
Solving... 5.316 seconds

Find the answer above or in the /mnt/data/sudoku2 file!
```

For several files:

```
lanfeust@troy ~/dev/tests docker run --rm -v "$(pwd)/data:/mnt/data" feust/sudoku-solver:2.1 data/$(cd data && ls)

=== Reading file /mnt/data/data/sudoku2_solved.txt ===
File doesn't exist or isn't a sudoku

=== Reading file /mnt/data/sudoku2.txt ===
[[1 7 8 5 4 6 2 3 9]
 [4 2 9 3 8 1 5 6 7]
 [5 6 3 9 2 7 1 8 4]
 [9 3 5 2 1 4 6 7 8]
 [7 4 1 8 6 5 9 2 3]
 [6 8 2 7 9 3 4 1 5]
 [2 5 6 4 7 8 3 9 1]
 [8 1 4 6 3 9 7 5 2]
 [3 9 7 1 5 2 8 4 6]]

Sudoku solving speed:
Parsing... 0.0001 seconds
Solving... 5.313 seconds

Find the answer above or in the /mnt/data/sudoku2 file!

=== Reading file /mnt/data/sudoku3 ===
[[8 1 6 5 4 2 9 3 7]
 [4 9 5 6 3 7 1 2 8]
 [7 3 2 1 9 8 6 4 5]
 [9 2 4 8 7 6 5 1 3]
 [5 8 7 3 1 9 4 6 2]
 [3 6 1 4 2 5 7 8 9]
 [6 4 8 7 5 3 2 9 1]
 [1 5 9 2 8 4 3 7 6]
```