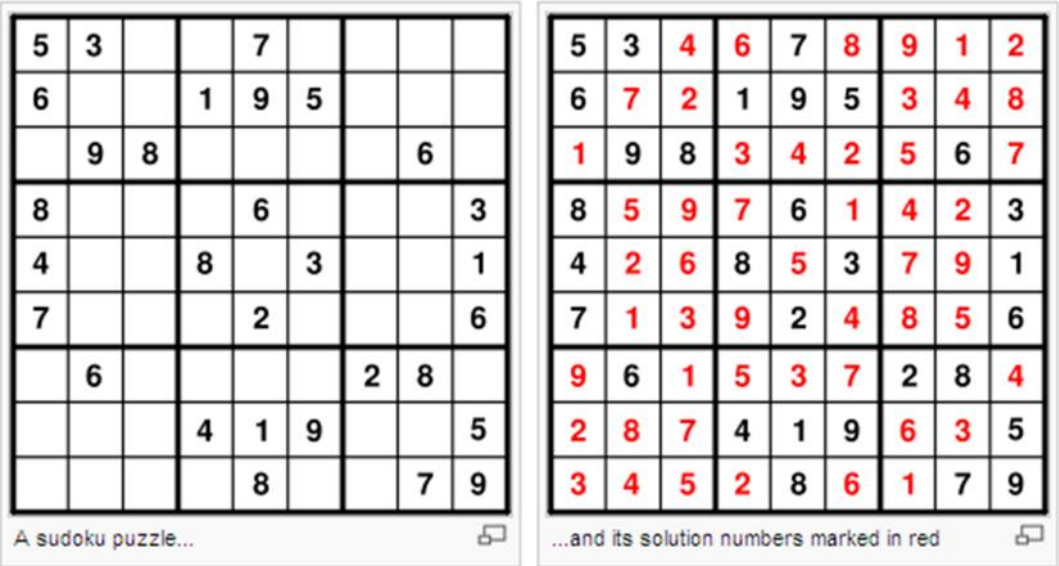
PROBLEM STATEMENT

This assignment consists of designing, implementing, analysing experimentally and writing a report on an *evolutionary algorithm* to solve *Sudoku puzzles*.

In a *Sudoku puzzle*, the objective is to fill a 9×9 grid with digits so that each column, each row, and each of the nine 3×3 subgrids that compose the grid contains all of the digits from 1 to 9, from an initial partially competed grid.



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

A sudoku puzzle...

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

...and its solution numbers marked in red

source: Wikipedia

An *evolutionary algorithm* is a general problem solving framework inspired to biological evolution. The pseudocode of an evolutionary algorithm is reported below.

```
BEGIN
  INITIALISE population with random candidate solutions;
  EVALUATE each candidate;
  REPEAT UNTIL ( TERMINATION CONDITION is satisfied ) DO
    1 SELECT parents;
    2 RECOMBINE pairs of parents;
    3 MUTATE the resulting offspring;
    4 EVALUATE new candidates;
    5 SELECT individuals for the next generation;
  OD
END
```

WHAT YOU NEED TO DO

1) **Design**:

1. Choose an appropriate solution space and solution representation.
2. Define an appropriate fitness function.
3. Define a crossover operator for the chosen representation.
4. Define a mutation operator for the chosen representation.
5. Decide how to initialise the population.
6. Decide selection and replacement methods.
7. Choose an appropriate termination criterion.

2) **Program**: implement an evolutionary algorithm in *Python* following the pseudocode given in the previous section and the problem-specific components above.

3) **Experiments**: run experiments for the three Sudoku grids provided on the ELE page, for population sizes 10, 100, 1000, 10000. Each experiment (i.e., a specific combination of grid and population size) needs to be ran 5 times (each one with a different random seed) and average performance across runs considered. In total these amount to 3 x 4 x 5 = 60 runs.

4) **Questions**: for the analysis of results, answer the following questions:

a. What population size was the best?
b. What do you think is the reason for your findings in question 8.a?
c. Which grid was the easiest and which the hardest to solve?
d. What do you think might be the reason for your findings in question 8.c?
e. What further experiments do you think it may be useful to do and why?

Write a report (maximum 4 A4 sides) covering design, experiments and questions. ***The report must have a separate section for each these. The section on design must cover separately each of the seven components. The section on questions must contain separate answers for each of the five questions.***