Programming assignment

Weight: 20%

Must Attempt: Yes

Notes:

- All program files must be compiled and tested on Turing before submission using the "submit" command.
- Programs that fail to compile on Turing will receive zero mark.
- Do not submit your program files in Moodle. They will not be marked.

This assessment relates to:

Learning Outcomes 1 2 3 4

Instructions

This assignment requires you to write a program to find the solution to a specific 9x9 grid ***Sudoku puzzle***.

To learn more about how Sudoku puzzles work see https://sudoku.com/how-to-play/sudoku-rules-for-complete-beginners/

Your task is to write a program that implements an AI technique, including search and structural techniques, which we have covered in the first five weeks of this unit to solve the puzzle as described above. You can write your program in Python, but you may write it in another language (if requested prior to submission). **The program has to be written so that it compiles and runs on *Turing*. If it doesn't compile on *Turing*, no marks will be given.**

I suggest removing or commenting out any code that is causing runtime errors and document any attempts at solving individual operations.

The source code should preferably be in one file. In addition, please submit a “readme” text file describing how to compile your program (python should be a single command). An optional third file that explains any command line arguments that we can use when running your program may be included. The name of the source code file should be in the form of your UNE username followed by the appropriate extension for the programming language used (like .py, .java, .c, or .cpp).

Your program should be terminal based and when runs, should provide all information about the author, what it does, a concise description of the method you have chosen to solve the puzzle, a list of parameters if they exist that can be used, etc. When your program is running, we expect intermediate feedback/output on what the program is doing. The last bit of output from your program should display the answer to the puzzle.

How you implement your solution and the program is up to you. However, the final mark will depend on a set of marking criteria including usability, information it provides, the ease of use, and appropriate feedback to the user, that is us who will be marking your submission. Please note: your program will be marked mostly from its output, but proper programming (including documentation) should be observed.

Grading scheme

The final mark of this programming assignment (out of a maximum of 100 marks) will be based on the following marking criteria:

Ease of use and understand: 15%

Run-time information: 20%

AI method used (i.e. methods that are not search or evolutionary algorithms will receive at most 50% of the mark on this criterion; hard-coding the solution in your program will be a straight zero): 30%

Concise description of the algorithm in the output: 10%

Clarity of presentation of the solution: 20%

Structure and documentation in code: 5%

Our Puzzle

The Sudoku puzzle you need to solve is depicted below. You can hardcode the initial number assignments as per the picture below. All further number assignments need to be valid for the rules of Sudoku and be generated by your algorithm as described above.

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