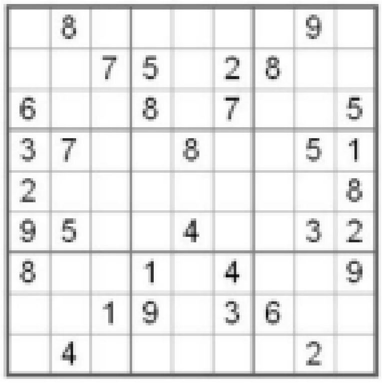
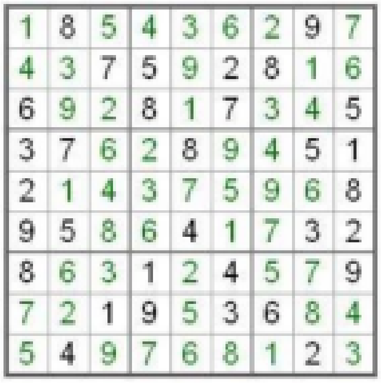
Problem Statement

Q1-Solve sudoku using Genetic Algorithm with crossover operation.

1. The objective is to fill a 9x9 grid with the digits 1 to 9, subject to the following constraints: each row, each column, and each of the nine 3x3 subgrids must contain a permutation of the digits from 1 to 9. The initial puzzle configuration provides a partially completed grid, such as the example grid illustrated below on the left. The Sudoku solver must fill in the remaining grid elements to obtain a solution like the one given on the right.



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



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

If you don't know about sudoku watch https://www.youtube.com/watch?v=kvU9_MVAiE0

Instructions:

1. The input to the problem is a partially completed grid, as shown in the figure above.
2. You need to populate the empty places in the grid with random digits to create one fully-populated individual state.
   a. You can improve this by looking at the board and using the information efficiently. So, instead of just generating random digits blindly, you can use the other digits in the board to be smart about how to populate the empty spaces in the grid.
   b. At the same time, don't overthink or over-do, then you will end up stuck.
3. (say) You want to start with N individual-states as your initial population. Then, repeat Step-2 'N' times to generate N different individuals.
4. Encode each individual in the population as a Chromosome/String.
   a. As mentioned in the book, pick an encoding strategy that makes it easier for the GA. This is one of the trickiest problems in GA.
   b. Remember, the book encodes individual-states in the 8-queens problem as a string of 8-digits. Each digit represents the row-no of each queen, starting from column 1.
5. One way to encode the sudoku board is as follows [1, 2, 3, 4, 5, 6, 7, 8, 9, 1, 2, 3, 4, 5, 6, 7, 8, 9, 1, 2, 3, 4, 5, 6, 7, 8, 9, 1, 2, 3, 4, 5, 6, 7, 8, 9, 1, 2, 3, 4, 5, 6, 7, 8,

    9, 1, 2, 3, 4, 5, 6, 7, 8, 9, 1, 2, 3, 4, 5, 6, 7, 8, 9, 1, 2, 3, 4, 5, 6, 7, 8, 9, 1, 2, 3, 4, 5, 6, 7, 8, 9] i.e. a list of 81 values concatenated row-wise. You could also choose to represent the board as a String of length 81. The choice is yours. Feel free to choose your own strategy and experiment as necessary.

    6. The simplest fitness function would use the number of duplicate numbers in the string/list. The lesser the duplicates the better the fitness. But, we recommend you to think and experiment with fitness functions, which could make your algorithm converge faster.
    7. There are several ways to select the crossover point. You may choose the mid-point to always be the crossover point, a few sudoku GA approaches recommend this (OR) choose the crossover point randomly, as shown in the book. The choice is yours.
    8. There are several approaches to determine when to terminate the GA program. You can stop when the average-fitness plateaus (OR) you can simply stop after some X iterations. After completion, print the 5 best individual-states from the final population.

2. Experiment and find out what improvements can help the algorithm to converge faster. For example, you can come up with a better fitness function or a better (heuristic) driven way to choose the crossover point. Likewise, you can see if there is a smarter way to favor one type of mutation over another..

Q2. Solve the following using Genetic algorithm:

z = x^2 – 2y + 3

(a) Find the minimum value that z, if x and y can only have integer values between 0 and 7.
(b) You can use a binary 3-bit representation for x and y. if x = 0, it would be represented by 000 and if x = 7 by 111. Likewise, the values for y would also require 3 bits. Hence, creating a suitable State-string/chromosome for this problem requires a 3 bit + 3 bit = 6 bit string representation (example: (say) x=1, y=1 The string is 001001) (c) You can choose to hardcode the equation and the constraints. But, a better solution is to use a library for equation parsing (example: https://pypi.org/project/Equation/). By using such a library, you can build a general GA solution which can work for any equation. Once the equation has been parsed successfully, you can ask the user to enter the range of values that the independent variables (ex: x,y) can accept.