Sudoku Solver

What is sudoku?

Sudoku is a logic-based puzzle. The objective of sudoku 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 contains all of the digits from 1 to 9.

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

How to solve sudoku?

In general, the user must fill all the cells as appropriate number to complete sudoku.

In programming there are many ways to solve sudoku. With our project we will ask the user to fill which cells she/he wants, and then our program going to complete the sudoku grid with numbers from 1-9 by

try every cell one by one until find the appropriate cell, after finding the right place it will check the 3×3 grid if is it completely right and done, if yes it will recur for other sub grids and do it again until finish the big 9×9 grid.

Functions used:

We used 7 functions to solve sudoku and the main method.

1- EntryGrid();
   This function will interact with user by asking questions and taking answers.

Also, it will initiate the grid as a 2-dimensional array, then using first FOR loop to loop through the 9 rows and the nested FOR loop will do the same with columns. To break the two nested FOR there is (else) with rows and columns Indicator= 10 which is bigger than FOR conditions, so will never pass to the loops.

2- IsInRow();

As we mention earlier sudoku must check that the number can not be twice at the same row and same column.

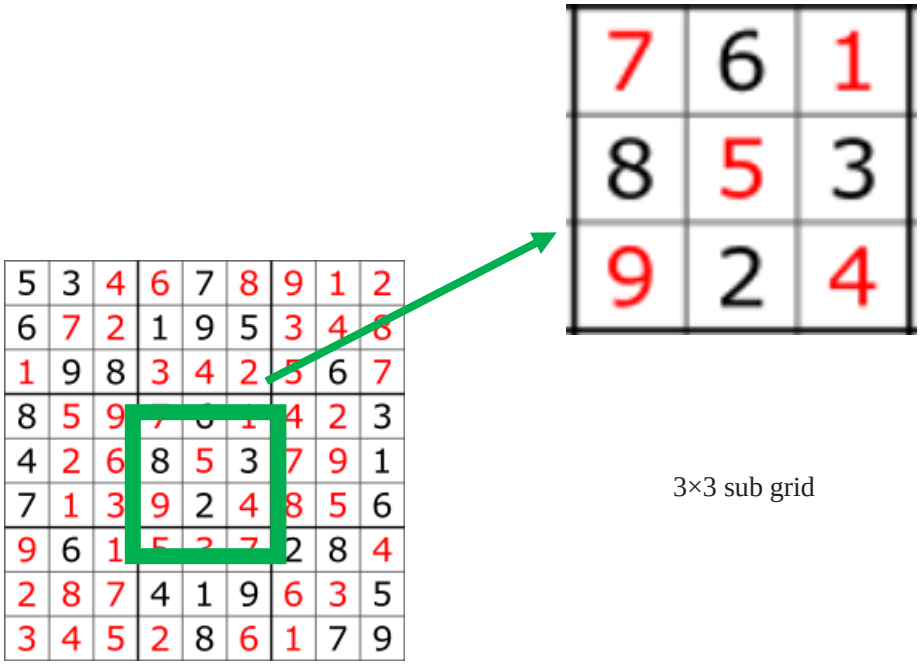
for loop: will check the row if any cell contains the same number, if yes return false.

3- IsInCol();

for loop: will check the column eight cells if any contains the same number, if yes return false.

4- IsInBox();

What we mean by this function? In sudoku, the big 9×9 grid consisting of nine 3×3 sub grids, each grid has a range of valid number from 1 to 9 without duplication as shown below.



9×9 grid

3×3 sub grid

First of all, the pointer needs to know from which cell start checking the grid! we can do that by creating a startrow and startcoulmns.

And then two nested FOR loops first one will check the row, a possible rows numbers are 3, and the same with second for wit columns. Let's assume we are targeting the grid start with cell [6,6] and walk throughout the process.

If condition will work as:

First iteration I = 0 ;	Second iteration I = 1;	Third iteration I = 2;
J =0;	J =0;	J =0;
[6+0 , 6+0] : [6,6]	[6+1 , 6+0] : [7,6]	[6+2 , 6+0] : [8,6]
J=1;	J=1;	J=1;
[6+0 , 6+1] : [6,7]	[6+1 , 6+1] : [7,7]	[6+2 , 6+1] : [8,7]
J =2;	J =2;	J =2;
[6+0 , 6+2] : [6,8]	[6+1 , 6+2] : [7,8]	[6+2 , 6+2] : [8,8]

5- IsValid();
   This function will make sure that all three functions above (IsInRow and IsInCol and IsInBox) are true so the cell is appropriate to be filled with the number.
6- solve();
   solve function will first make sure that the grid is not end yet. If not, it will going to check if column is last one which is 8^{th} it will be zero and next row so it can check the all coulmns and rows.
   Now, if the cell is not containing zero it is mean it contains a number from 1 to 9 and must be go back to the grid and increment column by one to next cell. If none of number is candidate then set the cell by zero and go back to find the appropriate number for this cell.
7- Print();
   It is all about showing the user how sudoku is solved.