SUDOKU SOLVER

/*------Created by – Doman Sarkar--------------------------------------

---------This program is made in C++ ------------------------------------*/

#SUDOKU SOLVER

This program helps solve Sudoku and even can be used to check if a Sudoku is error free or not. Input is done by entering the numbers and the vacant spaces could be entered by entering zero.

For e.g-

(user inputted array of 9x9 and empty spaces are replaced by zeros)

```
5, 4, 0, 0, 2, 0, 8, 0, 6
0, 1, 9, 0, 0, 7, 0, 0, 3
0, 0, 0, 3, 0, 0, 2, 1, 0
9, 0, 0, 4, 0, 5, 0, 2, 0
0, 0, 1, 0, 0, 0, 6, 0, 4
6, 0, 4, 0, 3, 2, 0, 8, 0
0, 6, 0, 0, 0, 0, 1, 9, 0
4, 0, 2, 0, 0, 9, 0, 0, 5
0, 9, 0, 0, 7, 0, 4, 0, 2
```

#Instructions

=> Make sure the inputted matrix should be 9x9 otherwise the code will not work properly.

=>If the code detects numbers not in the range 0-9 it will exit out of the program.

=>Enter 0 to represent the empty spaces in a Sudoku.

=>Output matrix of the solved Sudoku will be shown at the end and is the entered Sudoku can have more than one element in one place it will show them as a whole number in that place.

#Requirements

1.1-Hardware Requirements

1.1.1-RAM: 1GB (Minimum)

1.1.2-HDD: 2GB (Minimum)

1.1.3-Processor: Intel Celeron (Minimum)

1.2-Software Requirements

1.2.1-OS: Windows 10

1.2.2-Visual Studio Code or any IDE which supports C++

1.2.3-GCC 10.2 or latest version

#Modules

1-isSudokuSolved() – takes the grid as the parameter and it checks if the Sudoku is solved. If the Sudoku is solved then it will return true otherwise false.

2-disp() – takes grid as the parameter and it displays the 9x9 matrix.

3-remove() – removes the required element from a whole number greater than 9. For e.g- ‘n’=1347 and ‘k’=3 then resulting ‘n’ will be 147. Parameters are (address-of-n, element-which-we-have-to-remove).

4-dupGrid() – checks for duplicate value in a sub grid of 3x3 and it also removes it. Parameters are (address-of-grid, value-of-subgrid). Value-of-subgrid ranges from 0 to 8.

5-duplicate() – checks for duplicate values in a row and column and it also removes it. Parameter passed is the address of the grid.

6-missingInGrid() – takes (address-of-grid, value-of-subgrid) as parameters and fills the whole subgrid with the numbers which are missing in that subgrid. Value-of-subgrid ranges from 0to 8.

#####--------------For e.g-

Lets take the subgrid 6
0, 6, 0
4, 0, 2

0, 9, 0
In this subgrid the numbers missing are 1, 3, 5, 7, 8. The 0 places of the subgrid will be replaced by 13578 as a whole number.

Resulting subgrid will be:

```
13578,     6,      13578
4,         13578,          2
13578,      9,      13578
```

7-checkGrid() – takes the address of the grid as an parameter and checks the whole 9x9 matrix after the user inputted the matrix for abnormal values and terminates the program after printing the errors.

#Error handling

$ “entered no. is not between 0-9!” – This error occurs if user enters a number greater than 9 or less than 0

$ “empty row!” – This error occurs if the program finds the whole row empty or ‘0’.

$ “empty column!” – This error occurs if the program finds the whole column empty or ‘0’.

$ “duplicate number in same row!” – This error occurs if the program finds a number more than one time in a row.

$ “duplicate number in same column!” – This error occurs if the program finds a number more than one time in a column.