SUDOKU SOLVER

This Project aims to solve the sudoku given by user as an input.

Statement

Suppose we have a Sudoku grid and we have to solve this famous number maze problem, Sudoku. We know that Sudoku is a 9 x 9 number grid, and the whole grid are also divided into 3 x 3 boxes There are some rules to solve the Sudoku.

- We have to use digits 1 to 9 for solving this problem.
- One digit cannot be repeated in one row, one column or in one 3 x 3 box.

Technology Used

Visual Studio 2019

Installation

1- Check the system requirements for Visual Studio 2019.

2- Next, Download the Visual Studio 2019 from https://visualstudio.microsoft.com/downloads/.

3- Install and run the Visual Studio installer.

4- Next click on Continue.

5- Select Visual Studio Community 2019.

6- After the installer is installed, you can use it to customize your installation by selecting the feature sets or workloads that you want.

7- For C++, choose Desktop Development with C++.

8- Now, click on the Install button to begin the installation of Visual Studio 2019.

9- The installer will now download each component from the internet and starts the installation. This will take a while depending on your internet speed.

10- Once the installation is complete, you will be presented with the Installation succeeded message along with the option to register.

11- Now select Create a new project.

12- In the next page, select Console App. Note that you may have multiple variants of console apps (.NET Core, .NET Framework, etc). Select the one without parentheses. It should also have C++, Windows, Console tags.

13- Configure your project and continue. You can name the project however you'd like.

Working

Using backtracking algorithm, we will try to solve Sudoku problem. When some cell is filled with a digit, it checks whether it is valid or not. When it is not valid, it checks for other numbers. If all numbers are checked from 1-9, and no valid digit found to place, it backtracks to previous option.

Methods used

1- bool FindUnassignedLocation (int grid[N][N], int& row, int& col);

This function finds an entry in grid that is still unassigned. It searches the grid to find an entry that is still unassigned. If found, the reference parameters row, column will be set the location that is unassigned, and true is returned. If no unassigned entries remain, false is returned.

Parameters : grid(sudoku), row, column

Return Type : Bool(true or false)

2- bool isSafe (int grid[N][N], int row, int col, int number);

This function checks whether it will be legal to assign number to the given row, column. This function returns a boolean which indicates whether it will be legal to assign number to the given row, column location.

Parameters : grid(sudoku), row, column, number

Return Type : Bool(true or false)

3- bool isValidSudoku (int grid[ ][N]);

This function checks if the solution of sudoku puzzle is valid or not if the sudoku given as an input is completely filled.

Parameters : grid(sudoku)

Return Type : Bool(true or false)

4- bool SolveSudoku (int grid[N][N]);

This function takes a partially filled-in grid and attempts to assign values to all unassigned locations in such a way to meet the requirements for Sudoku solution (non-duplication across rows, columns and boxes).

Parameters : grid(sudoku)

Return Type : Bool(true or false)

5- bool UsedInRow (int grid[N][N], int row, int number);

This function Returns a boolean which indicates whether an assigned entry in the specified row matches the given number.

Parameters : grid(sudoku), row, number

Return Type : Bool(true or false)

6- bool UsedInCol (int grid[N][N], int col, int number);

This function returns a boolean which indicates whether an assigned entry in the specified column matches the given number.

Parameters : grid(sudoku), column, number

Return Type : Bool(true or false)

7- bool UsedInBox (int grid[N][N], int boxStartRow, int boxStartCol, int number);

This function returns a boolean which indicates whether an assigned entry within the specified 3x3 box matches the given number.

Parameters : grid(sudoku), box starting row, box starting column, number

Return Type : Bool(true or false)

8- void printGrid (int grid[N][N]);

A utility function to print grid or the sudoku.

9- Driver Function

In this driver function, following steps are followed:-

1- Take an input from user and assigned that input into 9x9 grid and unassigned location as ‘0’ (zero).

2- Check whether the number given as input is between 0 to 9 including 0 and 9.If it is not that print the appropriate message and terminate the program otherwise continue.

3- Check if the sudoku is completely filled or not.

4- If the sudoku is completely filled then check whether the sudoku is valid or not. If valid then print sudoku with appropriate message. If it is invalid then print appropriate message and terminate the program.

5- If the sudoku is partially filled then solve the sudoku. If the solution of sudoku exists then print the solved sudoku as an output. If the solution of sudoku is not possible or doesn’t exist then print the appropriate message and terminate the program.

10- Build and run the project.

1- Select Build Solution from the Build menu.

2- To run the code, on the menu bar, choose Debug->Start without Debugging.

11- Now run this program and try different type of sudoku as an input.