**A PROJECT REPORT ON:**

**“SUDOKO GAME USING C++”**

**SUBMITTED BY:**

**SAYMA NAAZ**

**C++PROGRAMMER INTERN**

**SUBMITTED TO:**

**CODESPEEDY TECHNOLOGY PVT LTD**

**TABLE OF CONTENTS:**

1. *SHORT DESCRIPTION*
2. *ABOUT*
3. *CODE FOR SUDOKU CHECKER*
4. *OUTPUT OF SUDOKU CHECKER*
5. *CODE FOR SUDOKU SOLVER*
6. *OUTPUT OF SUDOKU SOLVER*
7. *COMPLEXITY ANALYSIS*
8. *CONCLUSION*
9. *REFERENCE*

**SHORT DESCRIPTION:**

Sudoku is logic-based combinatorial number-placement puzzle. The Objective of this game is to fill 9X9 grid with digits 1 to 9 such that the digit should not repeat in the row, column and in the sub-grid of 3X3.

In this packet the C++ programming language is used to solve the Sudoku game. In the Sudoku Checker program we have to check whether the unsolved sudoku is valid to solve or not and then we apply algorithm to the unsolved sudoku which gives the output a solved sudoku in Sudoku Solver program.

**ABOUT :**

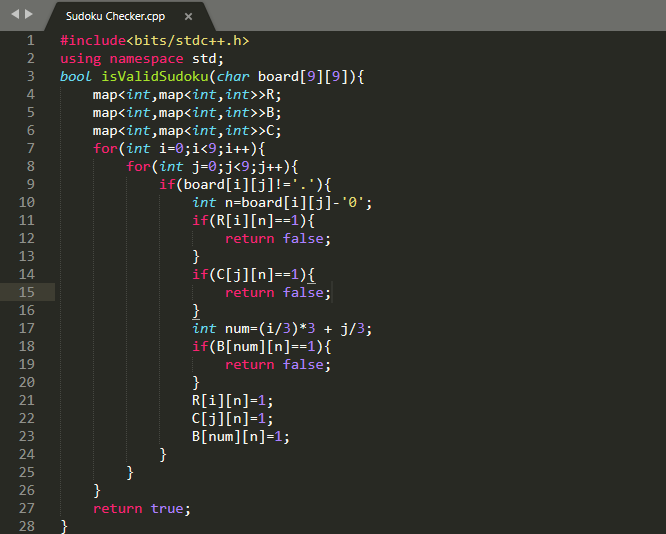
To solve the sudoku we use recursion with backtracking. Backtracking is useful algorithm for solving problem with recursion by building a solution incrementally. Generally, backtracking involves starting with a possible solution, and if it doesn’t work, you backtrack and try another solution until you find the correct one.

Backtracking is useful for the following three types of problems:

1. Decision-making problems to find a suitable solution to a problem.
2. The best solution to optimize the algorithm.
3. Count problems to find the suitable solution to a problem.

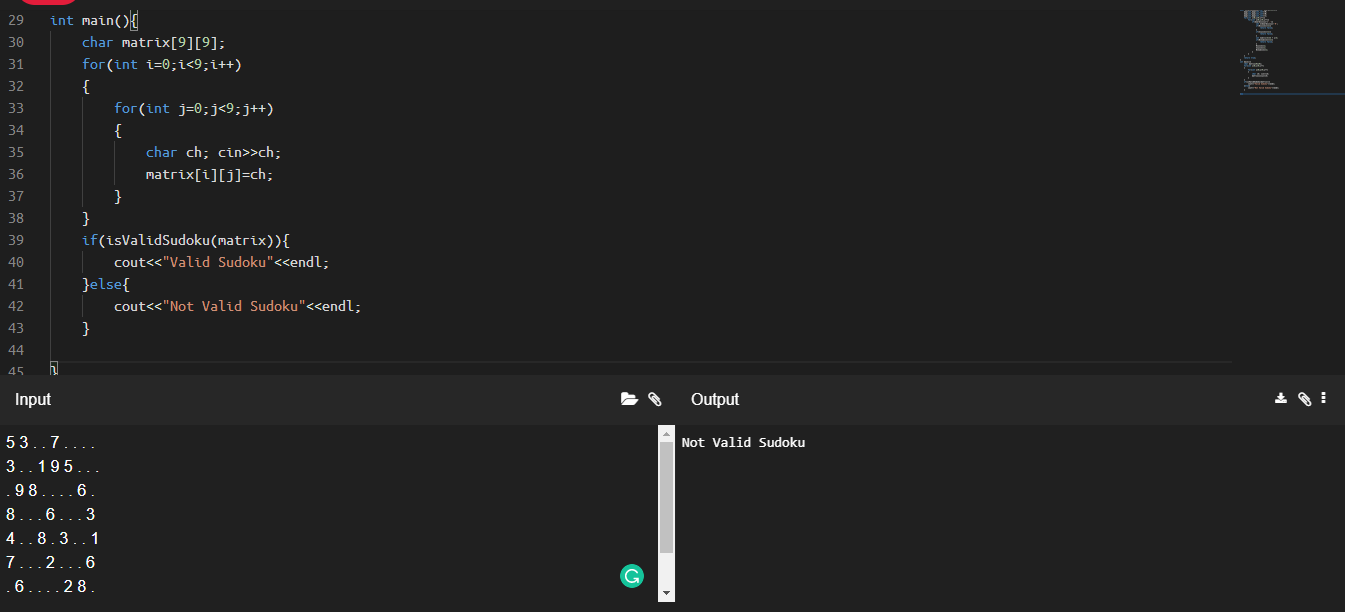
To solve sudoku we use C++ programming. In the program in the main function we initialize the sudoku matrix of size 9X9 and taking input the sudoku matrix. To solve the sudoku we define a function named sudoku solver, in this function we recursively call the function to solve the sudoku and at every cell if a digit is correct to place or not, to check this we define a function named Can Place function, and in this function we check if the digit is occurred in row, column, or in 3X3 matrix(sub-matrix). If Can Place function returns true and then sudoku solver also returns true then we print the solved sudoku matrix as output otherwise print “Sudoku cannot be solved”. And if at any point we are unable to place any digit then we backtrack and remove the previously filled digit and try another digit at that place and moves ahead.

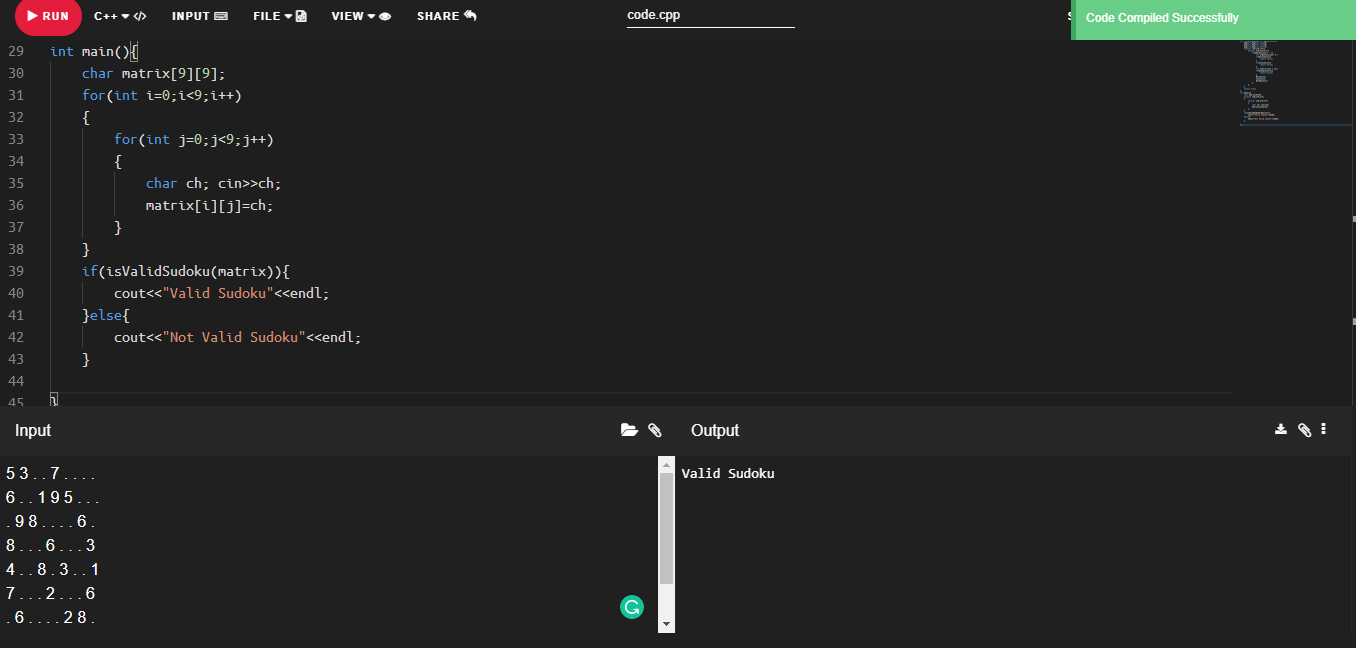
CODE FOR SUDOKU CHECKER:



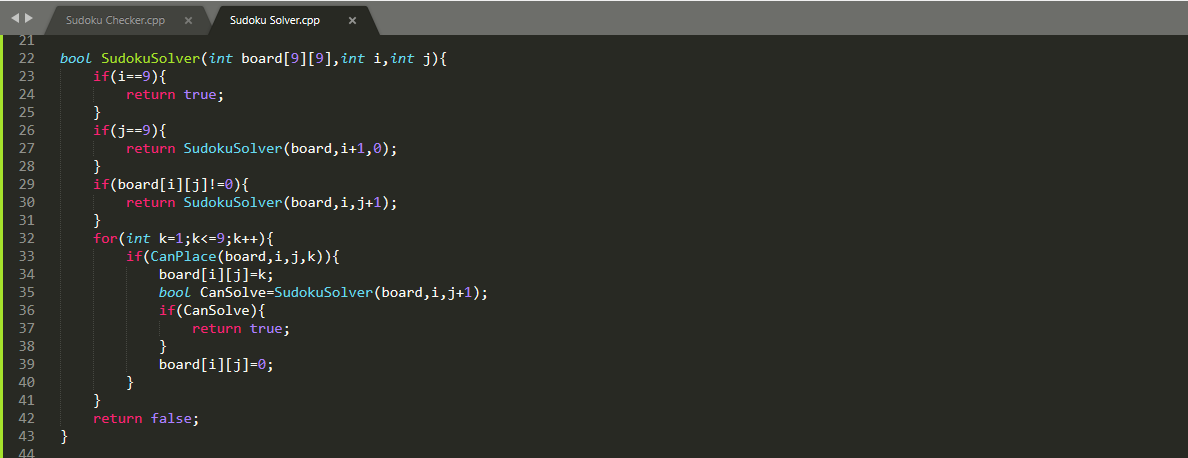
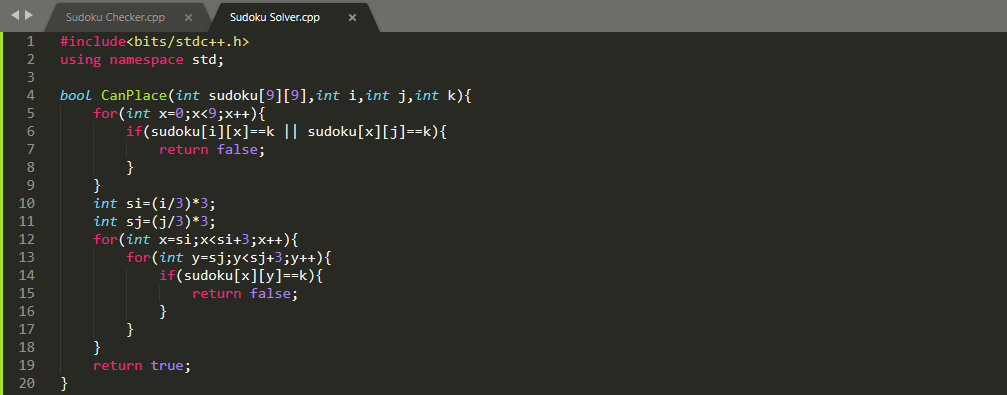


**OUTPUT OF SUDOKU CHECKER:**

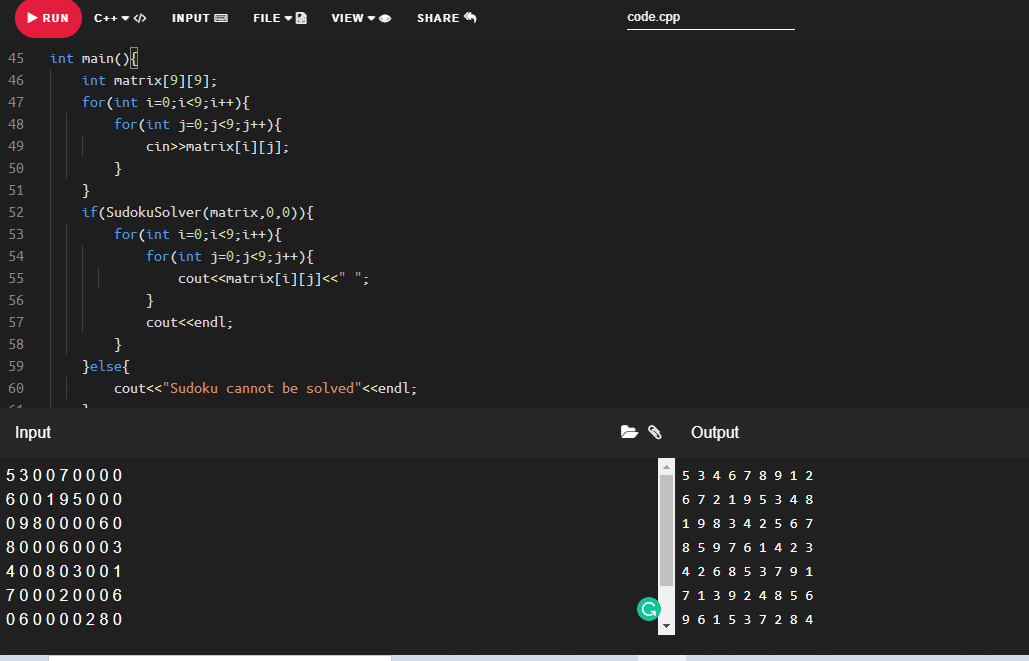
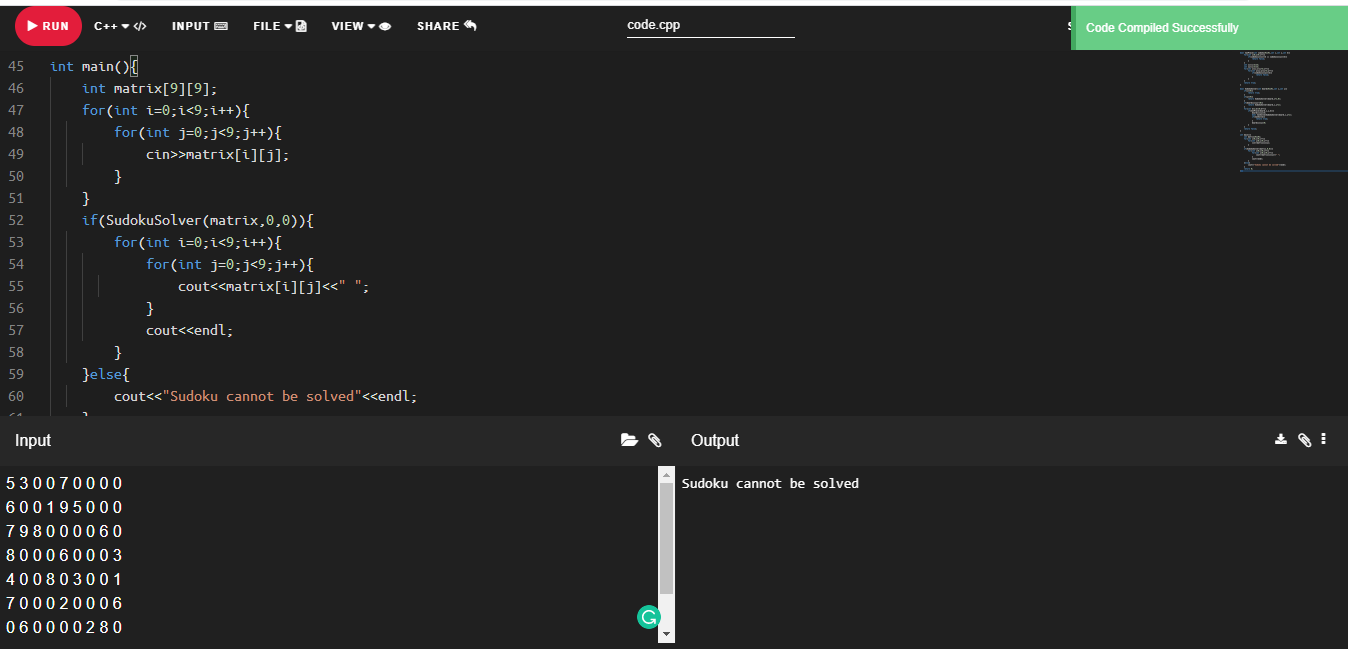
****

****

**CODE FOR SUDOKU SOLVER:**

****

**OUTPUT OF SUDOKU SOLVER:**

****

**COMPLEXITY ANALYSIS:**

***Space Complexity: O(m),***

***Time Complexity: O(9^m),***

***m- number of spaces to be filled.***

**CONCLUSION:**

To solve the sudoku recursion with backtracking has been used which tries the all possible solutions and finds the best possible solution which solves the 9X9 sudoku and time and space complexity has noted.