

# PROJECT REPORT

A report submitted in partial fulfillment of the requirements for the



# Project on

**“ SUDUKO SOLVER ”**

School of Computer Science & Engineering

By

**HARSH M DHAROD**

22SC114281011 11

**AMEYA P AMANAGI**

22SC114281009 09

**AMEY V KURADE**

22SC114281013 12

**ZAID A JAMADAR**

22SC114281022 20

Program: Bachelor of Computer Science and Engineering Class: FY BTech

Under Supervision of

## Mrs. Veena. Mali

Academic Year: 2022-2023



School of Computer Science & Engineering



# CERTIFICATE

This is to certify that the “**Project Report**”

On

## “ SUDUKO SOLVER ”

submitted by

**HARSH M DHAROD**

22SC114281011 11

**AMEYA P AMANAGI**

22SC114281009 09

**AMEY V KURADE**

22SC114281013 12

**ZAID A JAMADAR**

22SC114281022 20

Program: B Tech Class: FY B. Tech (Div. A)

is work done by him/her and submitted during the 2022 – 2023 academic year, in partial fulfillment of the **Project.**

**Sanjay Ghodawat University, Kolhapur**

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| **Mrs. Veena. Mali** | **Ms. Deepika Patil** | **Dr. B. Suresh Kumar** | **Examiner** |
| **Project Guide** | **PBL Co-Ordinator** | **Head, SOCSE** |  |



## DECLARATION

I the undersigned solemnly declare that the report of the project work entitled **“ SUDUKO SOLVER ”** which is carried out under the supervision of **Mrs. Veena. Mali** I assert that the statements made and conclusions drawn are an outcome of the project work. I further declare that to the best of my knowledge and belief that the project report does not contain any part of any work which has been submitted for the award of any other degree/diploma/certificate in this University or any other University.

**Student Name :**

**HARSH M DHAROD**

22SC114281011 11

**AMEYA P AMANAGI**

22SC114281009 09

**AMEY V KURADE**

22SC114281013 12

**ZAID A JAMADAR**

22SC114281022 20

**Class:** FY B Tech (Div. A)

**Sanjay Ghodawat University ,Kolhapu r**



## ACKNOWLEDGMENT

First, I would like to thank my Head of the School **Dr. B. Suresh Kumar** for constructive criticism throughout my project. I would like to thank PBL coordinator **Mrs. Deepika Patil** and Department Project Guide **Mrs. Veena. Mali** for support and advices to get and complete internship in above said organization. It is indeed with a great sense of pleasure and immense sense of gratitude that I acknowledge the help of these individuals. I am extremely grateful to my department staff members and friends who helped me in successful completion of this project.

**Sanjay Ghodawat University ,Kolhapu r**



## ABSTRACT

Sudoku is a popular logic-based puzzle game that challenges players to fill a 9\*9 grid with numbers from 1 to 9 ,ensuring that each row, column , and 3\*3 sub-grid contains every digit exactly once. This abstract presents a sudoku solver development for digital systems. The solver employees a combination of advanced algorithms & data structures to efficiently solve sudoku puzzle of varying complexities. By utilizing constraint propagation techniques and a **Backtracking algorithm**, the solver systematically explores possible solutions, ensuring accuracy and completeness. The system has been implemented and extensively tested , demonstrating its effectiveness in solving sudoku puzzles accurately and in a **timely** **manner**.

The solvers modular design allows for easy integration into digital platforms, making it a valuable tool for sudoku enthusiasts and developers looking to incorporate sudoku-solving capabilities into their projects. The efficiency and performance of the solver have been evaluated and compared to existing state-of-the-art solvers , showcasing its competitive advantages. Overall , the sudoku solver presented in this **project offers a robust and efficient solution** for sudoku puzzles in digital system.

**Sanjay Ghodawat University ,Kolhapu r**

## TABLE OF CONTENT’S

|  |  |  |
| --- | --- | --- |
| **SR.NO** | **Title** | **Page No.** |
| 1 | Introduction | 01-04 |
| 2. | Objective | 05 |
| 3 | System Requirements Specification (SRS) | 06 |
| 4 | Methodology | 07-08 |
| 5 | Implementation | 09-14 |
| 6 | Result | 15-16 |
| 7 | Conclusion And Future Scope | 17-18 |
| 8 | References | 19 |

**Sanjay Ghodawat University ,Kolhapu r**

# INTRODUCTION

• **INTRODUCTION :**

Sudoku Solver is an algorithmic program designed to solve Sudoku puzzles using a backtracking algorithm. Sudoku is a popular puzzle game that requires logical reasoning and problem-solving skills. It consists of a 9x9 grid that is divided into nine 3x3 sub-grids. The objective of the game is to fill in each cell of the grid with a number from 1 to 9, such that each row, column, and sub-grid contains the numbers 1 to 9 only once.

Sudoku Solver is a program that can solve any valid Sudoku puzzle in a matter of seconds. The program takes a Sudoku puzzle as input and outputs the solution to the console or a file. The Sudoku puzzle can be entered manually or read from a file. The program checks the input for errors and invalid input to ensure the puzzle is solvable.

The algorithm used by Sudoku Solver is a backtracking algorithm. This algorithm is a bruteforce approach that tries every possible combination until a solution is found. The algorithm starts at the first empty cell and tries all numbers from 1 to 9. If a number is valid, the algorithm moves to the next cell. If no number is valid, the algorithm backtracks to the previous cell and tries the next number. This process continues until a solution is found.

To represent the Sudoku board in C language, Sudoku Solver uses a two-dimensional array or a struct with rows, columns, and cells. Each cell can contain a number between 1 and 9 or be empty (represented as 0).

Sudoku Solver provides a reliable and user-friendly tool for Sudoku enthusiasts and players.

**Sanjay Godawat University, Kolahpur** 1 **F.Y B.Tech SoCSE**

* **PROBLEM DEFINITION :**

The **SUDUKO SOLVER** is done manually by manpower for adding the valid questions . A certain based system could identify the organization to utilize its resource better.

The current problems are :

* Manual operator control.
* Lots of paper work to solve when done manually.
* Inability of modification of data in output.
* Wastage of paper
* No option to save

Henceforth ,based on the given problems and information the SUDUKO SOLVER can be more efficient.

* **SCOPE :**

1. The objectives of the proposed Project is to increase the Thinking Capability.
2. The Game having all the records which u perform in playing you can Select Easy, hard level according to your choice. You can make your own Sudoku and at any Step you can go back to One Step as well as you can see the Solution of it.
3. It is manually a very difficult job to perform and its need a lot of recalling , reminding and mathematical calculation. The game of “Sudoku” helps to increase mental thinking , vision etc.

**Sanjay Godawat University, Kolahpur** 3 **F.Y B.Tech SoCSE**

• **PROBLEM IDENTIFICATION :**

1. You cannot save the game/solver.
2. Need to add questions manually every time.
3. You cannot play it as it only solves the given valid question .

# OBJECTIVES

1. The objective of Sudoku Solver is to provide a program that can solve any valid Sudoku puzzle efficiently and accurately using a backtracking algorithm.
2. The program is designed to take a Sudoku puzzle as input and output the solution to the console or a file. Sudoku Solver also provides input validation, error handling, and optimization for a reliable and user-friendly tool for Sudoku enthusiasts and players.
3. The program aims to enhance the user experience by making the process of solving Sudoku puzzles easier and more enjoyable.
4. Ultimately, the objective of Sudoku Solver is to provide a valuable tool for anyone who enjoys solving Sudoku puzzles.

# SYSTEM REQUIREMENTS SPECIFICATION

* **SOFTWARE REQUIREMENT**

* Turbo C
* Microsoft Visual Studio Code
* Dev C++

* **HARDWARE REQUIREMENT**

* Computer or laptop
* Intel(R) Core(TM) i3-Processor
* RAM-1 GB Minimum
* Storage-100G

# METHODOLOGY

* **ALGORITHM :**

STEP 1**.** Initialize an empty 9\*9 sudoku grid

STEP 2. Read the puzzle from input or use a pre-defined puzzle.

STEP 3. Find an empty cell in the grid.

STEP 4. Try each number from 1 to 9 in the cell.

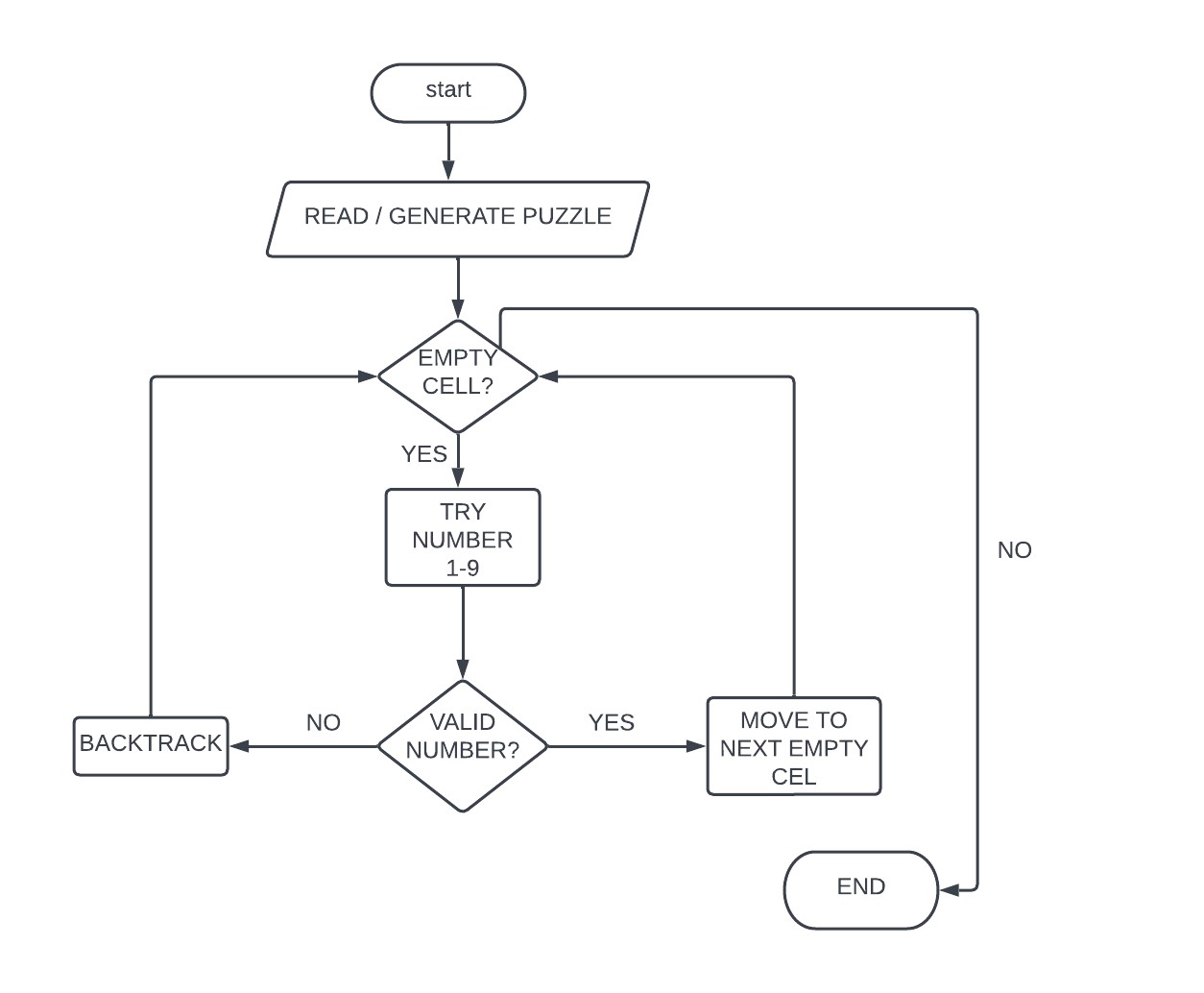
STEP 5. If a number is valid, move to the next empty cell.

STEP 6. If all numbers are invalid, backtrack to the previous cell & try a different number.

STEP 7. Repeat steps 3-6 until the entire grid is filled or a solution is find.

**Note: In STEP 4, a number considered valid if it doesn’t already exist in a same row, column, or 3\*3 sub-grid as the cell being filled.**

* **FLOW DIAGRAM (Flow Chart) :**



# IMPLEMENTATION

The code includes the necessary header files: stdio.h for standard input/output functions and dos.h for color manipulation (used later in the code).

Two macros are defined using escape sequences to set the text color in the console. CYAN sets the text color to cyan, and RESET resets the text color to default.

The function isAvailable checks if a number num can be placed in the specified position (row, col) in the Sudoku grid. It checks if the number exists in the same row, column, or 3x3 subgrid. If the number is available, it returns 1; otherwise, it returns 0.

The function fillSudoku Is a recursive function that tries to fill the Sudoku grid. It takes the Sudoku puzzle, the current row, and the current column as parameters. The base case for the recursion is when both the row and column values exceed their limits (9x9 grid). In this case, it returns 1 to indicate a successful solution.

Inside the fillSudoku function, it checks if the current position already contains a non-zero value. If so, it moves to the next position (column-wise or row-wise) and continues the recursion.

If the current position is empty (contains 0), it iterates from 1 to 9 and checks if each number can be placed at the current position using the isAvailable function. If a number is available, it places that number at the current position, updates the row and column, and makes a recursive call to fillSudoku for the next position.

If a solution is found in any of the recursive calls, it returns 1 to indicate a successful solution. If no solution is found, it backtracks by setting the current position to 0 and tries the next number.

The main function Initializes a 9x9 Sudoku puzzle with initial values. It displays the given question puzzle and prompts the user to enter a valid option.

Based on the user’s input, the code executes the corresponding case. In case 1, it calls the fillSudoku function with the initial position (0, 0) and checks if a solution exists. If a solution is found, it displays the solved Sudoku grid using nested loops. It also changes the console color to light cyan using system(“color 0B”) and prints a success message.

If the user enters an invalid option, it displays an “INVALID” message.

Finally, the main function returns 0 to indicate successful program execution.

**F.Y B.Tech**



**“INTRODUCTION OF C PROGRAMMING”**

**C** is a general-purpose programming language that is extremely popular, simple, and flexible to use. It is a structured programming language that is machine-independent and extensively used to write various applications, Operating Systems like Windows, and many other complex programs like Oracle database, Git, Python interpreter, and more.

It is said that ‘C’ is a god’s programming language. One can say, C is a base for the programming. If you know ‘C,’ you can easily grasp the knowledge of the other programming languages that uses the concept of ‘C’

In 1972, a great computer scientist Dennis Ritchie created a new programming language called ‘C’ at the Bell Laboratories. It was created from ‘ALGOL’, ‘BCPL’ and ‘B’ programming languages. ‘C’ programming language contains all the features of these languages and many more additional concepts that make it unique from other languages.

‘C’ is a powerful programming language which is strongly associated with the UNIX operating system. Even most of the UNIX operating system is coded in ‘C’. Initially ‘C’ programming was limited to the UNIX operating system, but as it started spreading around the world, it became commercial, and many compilers were released for cross-platform systems. Today ‘C’ runs under a variety of operating systems and hardware platforms. As it started evolving many different versions of the language were released. At times it became difficult for the developers to keep up with the latest version as the systems were running under the older versions. To assure that ‘C’ language will remain standard, American National Standards Institute (ANSI) defined a commercial standard for ‘C’ language in 1989. Later, it was approved by the International Standards Organization (ISO) in 1990. ‘C’ programming language is also called as ‘ANSI C’.

**Why Learning C ?**

Learning ‘C’ as the main language will play an important role while studying other programming languages. It shares the same concepts such as data types, operators, control statements and many more. ‘C’ can be used widely in various applications. It is a simple language and provides faster execution. There are many jobs available for a ‘C’ developer in the current market.

‘C’ contains 32 keywords, various data types and a set of powerful built-in functions that make programming very efficient.

**F.Y B.Tech**

Another feature of ‘C’ programming is that it can extend itself. ‘C’ program contains various functions which are part of a library. We can add our features and functions to the library. We can access and use these functions anytime we want in our program.

**ADVANTAGES OF LEARNING C** :

1. **It is easy to understand :** One of the main reasons why people choose C over other [**programming languages**](https://unstop.com/blog/best-programming-languages) is its simplicity. C is a highly portable language as programs coded in it are far more fast and efficient. This makes learning C easier than any other programming language. You can easily grasp the concepts behind C because there aren't many keywords or symbols involved.
2. **Presence of many Libraries :** C Language provides lots of built-in functions which consist of system-generated functions and user-defined functions. Many general functions can be used to develop a program, while the programmer can also create a function as per their requirements, which is called a user-generated/defined function, in C Compiler.
3. **Easy to write :** Another reason why C is so popular as an efficient language among programmers is that it allows them to create their own software without having to worry about syntax errors. If you're not familiar with coding, then using structured language C will help you develop better skills.
4. **Portable :** Since C is based on ASCII characters, it works well across different platforms including Windows, Linux, Mac OS X, Android, iOS, etc. Therefore, you can run your C programs anywhere regardless of where you live.
5. **Low Cost** : If you want to build something from scratch, then C is definitely worth considering because of its simple structure, you won’t spend much time trying to figure out whether you’ve made a mistake or not when developing a program. And if you decide someone else for your task then they would charge less .

**F.Y B.Tech**

**SOURCE CODE :**

#include <stdio.h>

#include<dos.h>

#define CYAN "\e[0;106m"

#define RESET "\e[0;106m"

int isAvailable(int puzzle[9][9], int row, int col, int num)

{

int rowStart = (row/3) \* 3;

int colStart = (col/3) \* 3; int i, j;

for(i=0; i<9;i++)

{

if (puzzle[row][i] == num)

return 0;

if (puzzle[i][col] == num)

return 0;

if (puzzle[rowStart + (i%3)][colStart + (i/3)] == num)

return 0;

} return 1;

} int fillSudoku(int puzzle[9][9], int row, int col)

{

int i;

if(row<9 && col<9)

{

if(puzzle[row][col] != 0)

{ if((col+1)<9)

return fillSudoku(puzzle, row, col+1);

else if((row+1)<9)

return fillSudoku(puzzle, row+1, 0); else

return 1;

} else {

for(i=0; i<9; ++i)

{

if(isAvailable(puzzle, row, col, i+1))

{

puzzle[row][col] = i+1; if((col+1)<9)

{

if(fillSudoku(puzzle, row, col +1))

return 1;

else puzzle[row][col] = 0;

}

else if((row+1)<9)

{

if(fillSudoku(puzzle, row+1, 0))

return 1;

else puzzle[row][col] = 0;

} else return 1;

}

} } return 0; } else return 1;

int main()

{

int i, j;

int PUZZLE;

int puzzle[9][9]={{0, 0, 0, 0, 0, 0, 0, 9, 0},

{1, 9, 0, 4, 7, 0, 6, 0, 8},

{0, 5, 2, 8, 1, 9, 4, 0, 7},

{2, 0, 0, 0, 4, 8, 0, 0, 0},

{0, 0, 9, 0, 0, 0, 5, 0, 0},

{0, 0, 0, 7, 5, 0, 0, 0, 9},

{9, 0, 7, 3, 6, 4, 1, 8, 0},

{5, 0, 6, 0, 8, 1, 0, 7, 4},

{0, 8, 0, 0, 0, 0, 0, 0, 0}};

printf("THE GIVEN QUESTION :\n"); printf("{0, 0, 0, 0, 0, 0, 0, 9, 0}\n"); printf("{1, 9, 0, 4, 7, 0, 6, 0, 8}\n"); printf("{0, 5, 2, 8, 1, 9, 4, 0, 7}\n"); printf("{2, 0, 0, 0, 4, 8, 0, 0, 0}\n"); printf("{0, 0, 9, 0, 0, 0, 5, 0, 0}\n"); printf("{0, 0, 0, 7, 5, 0, 0, 0, 9}\n"); printf("{9, 0, 7, 3, 6, 4, 1, 8, 0}\n"); printf("{5, 0, 6, 0, 8, 1, 0, 7, 4}\n");

printf("{0, 8, 0, 0, 0, 0, 0, 0, 0}\n\n");

printf("ENTER THE VALID OPTION :\n"); scanf("%d",&PUZZLE);

printf("\n+-----+-----+-----+\n");

switch(PUZZLE)

{

case 1:if(fillSudoku(puzzle, 0, 0))

{

for(i=1; i<10; i++)

{

for(j=1; j<10;j++)

printf("|%d", puzzle[i-1][j-1]);

printf("|\n");

if (i%3 == 0) printf("+-----+-----+-----+\n");

}

system("color 0B"); printf("\n YOUR REQUIRED SOLUTION !!"); printf(" \n THANK YOU ");

}break;

default :

printf("INVALID ");

}

return 0;

}

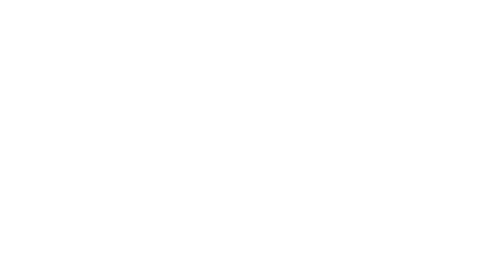
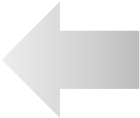
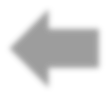
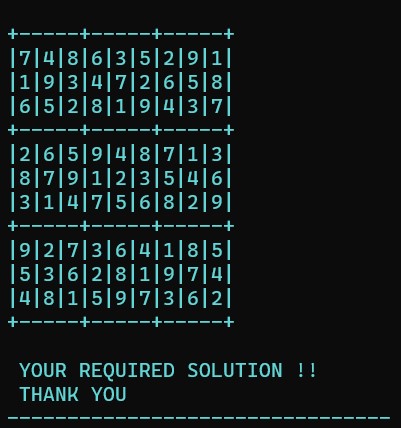
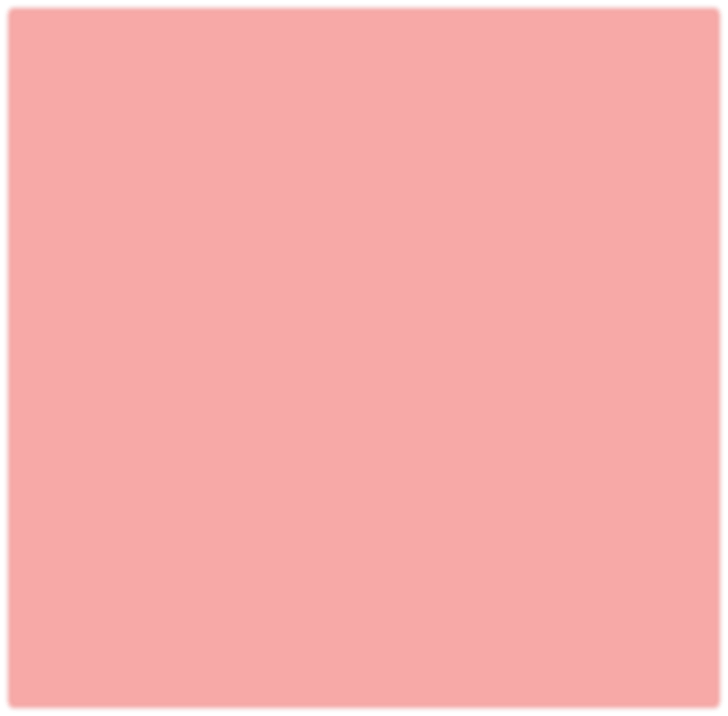
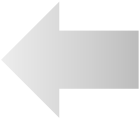
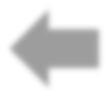
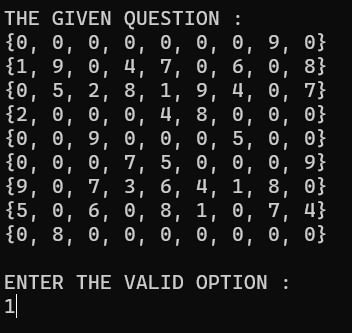
# RESULT

|  |
| --- |
| The side picture shows the inserted question and asks for  the valid option to continue |

**1.**

.

**2.**



This picture shows the final result

of the project after entering the

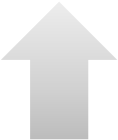
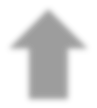
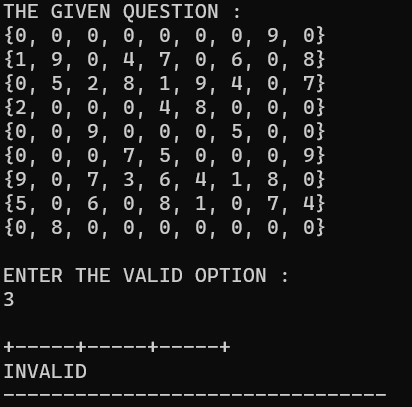
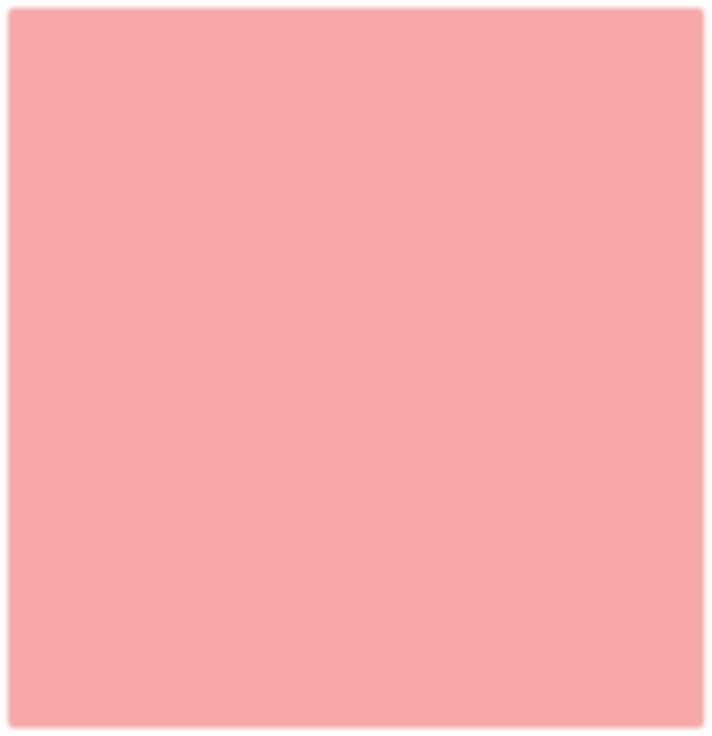
valid option. And automatically

changes

the text to colored format

as for new experience .

**3.**



Now, if the user inserts invalid option then he/she won’t get the solved answer of the question ,and hence displays

“INVALID”.

# CONCLUSION & FUTURE SCOPE

• **FUTURE SCOPE :**

These are some of the main future scopes for improvement.

1. Instant help by giving hints (if required) and users progress while playing.
2. Score system based on time and accuracy, and database to keep track of top ten record.
3. We will add 3 modes of level playing which would include Easy , Medium & Hard.
4. After giving time limit to each difficulty mode ,it will help to improve build the user’s logic.

**Sanjay Godawat University, Kolahpur** 17 **F.Y B.Tech SoCSE**

• **CONCLUSION :**

This was the project of system design about “ **SUDOKO SOLVER** ”.

Development of this system takes a lot of efforts . This study has show the pencil-and-paper algorithm is a feasible method to solve any Sudoku puzzle. The algorithm is also an appropriate method to find a solution faster and more efficient compared to brute force algorithm.

Though every task is never said to be perfect in the development field and even more improvement might be possible in this system.

Me and my team gained a lot of knowledge about this project. We understood how the system works what are the library functions, **Backtracking** and all. Hence we tried our best and put all the efforts for this project. Also in future we will add more and more options for users so they feel more happy and satisfied . Hence, sudoku solver is a brain game and to solve high difficulty level we can you this program. It was very nice project we came across with its outcomes**.**

**Sanjay Godawat University, Kolahpur** 18 **F.Y B.Tech SoCSE**

# REFERENCES

**WEBSITES :**

1. [**http://en.wikipedia.org/wiki/sudoku**](http://en.wikipedia.org/wiki/sudoku)
2. [**http://sudoku.sourceforge.net/**](http://sudoku.sourceforge.net/)
3. [**https://www.geeksforgeeks.org/suduko-backtracking-7/amp/**](https://www.geeksforgeeks.org/suduko-backtracking-7/amp/)
4. [**https://youtu.be/f\_5FgfvHw30**](https://youtu.be/f_5FgfvHw30)
5. [**https://codereview.stackexchange.com/questions/37430/suduko-solver-in-c**](https://codereview.stackexchange.com/questions/37430/suduko-solver-in-c)

**BOOKS :**

1. **Devlin Keith : “The Numbers Game”**
2. **Data Structure using C**

**Sanjay Godawat University, Kolahpur** 19 **F.Y B.Tech SoCSE**