Project Report: Sudoku Generator and Solver

Introduction:

The Sudoku Solver and Generator project aims to create a program that can generate random Sudoku puzzles of varying difficulty levels and provide solutions for them. Sudoku is a logic-based number placement puzzle where the objective is to fill a 9x9 grid with digits so that each column, each row, and each of the nine 3x3 subgrids contains all of the digits from 1 to 9 without repetition.

Objective:

The main objectives of this project are:

-Generate random Sudoku puzzles.
-Provide functionalities to solve Sudoku puzzles manually or automatically.
-Implement varying difficulty levels for the generated puzzles.
-Allow users to give up and see the solution after a certain time.

Implementation Process:

The project is implemented in C programming language using standard libraries such as stdio.h, stdlib.h, time.h, curses.h, and unistd.h. Here is a brief overview of the implementation process:

1.Generating Sudoku: The program generates a random Sudoku grid by filling the diagonal cells with random numbers and then using a backtracking algorithm to solve the puzzle.

2.User Interface: The program provides a command-line interface where users can choose to solve the Sudoku puzzle manually or automatically. It also includes a timer feature to track the elapsed time.

Difficulty Levels: Users can select the difficulty level of the Sudoku puzzle, ranging from easy to very hard, based on the number of hints provided.

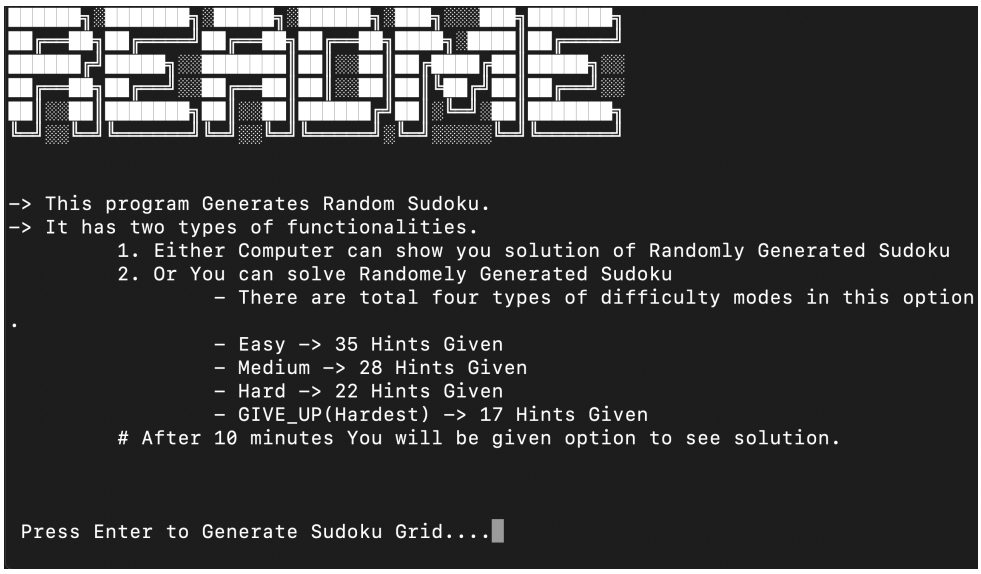
Solving Sudoku: The program allows users to solve the Sudoku puzzle manually by entering digits for empty cells. It checks the validity of the input and updates the grid accordingly.

Automated Solution: Users can also choose to see the solution automatically after a certain time if they are unable to solve the puzzle.

Contribution by Members:
The project was developed collaboratively, with each member contributing to different aspects:

Deep: Developed the user interface using curses library and handled input/output functionalities and helped interconnect/develope all the function in main programme.
Shivam: Implemented the Sudoku generation algorithm and Adding the timer functionality to track elapsed time and prompt users to see the solution if they exceed the time limit.
Devansh: Solver function of sudoku.
Raghav: Testing and debugging the program to ensure functionality and reliability.

ScreenShots –




Elapsed Time: 00:02
8 0 0 | 0 5 0 | 1 0 0
0 0 0 | 2 0 0 | 0 4 0
0 7 9 | 1 0 0 | 0 3 0
4 0 0 | 6 0 0 | 0 9 3
0 0 6 | 7 0 1 | 0 0 0
9 0 0 | 4 0 0 | 0 1 0
2 0 0 | 0 0 0 | 0 0 1
0 0 0 | 0 1 0 | 0 0 0
7 0 1 | 5 3 0 | 0 8 2
Generated Sudoku ^^^^
Hit enter to see solution Sudoku :