

Lab Report: sudoku solver Project

Author: Md. Rakib Hossain

Introduction:

The Sudoku solver project is a console-based application written in C, aimed at providing users with a Sudoku puzzle and allowing them to attempt solving it. The project utilizes a backtracking algorithm to find the solution to the Sudoku puzzle, and it includes user interface for Inputting the solution.

Objectives:

The main objective of this project is to implement a Sudoku solver that uses a recursive backtracking algorithm to find a solution to a given Sudoku puzzle. The project also includes a user interface for interaction, allowing users to input their solution and checking whether it is correct.

User Interface of the game:

```
WELCOME TO SUDOKU SOLVER

Instructions:

Sudoku is a popular number puzzle game that involves filling a 9x9 grid with numbers.
The grid is divided into nine 3x3 subgrids, and the objective is to fill each row, column, and subgrid with the numbers 1 through 9.
However, no number can be repeated in any row, column, or subgrid.

Let's start the play

Enter the difficulty level:
Ranging from 1 to 3
2

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

```
Provide your solution Sudoku (in a 2D matrix):
```

```
1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1
```

```
Your provided Sudoku solution isn't CORRECT!
```

```
Here is the solution of the Sudoku:
```

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

Implementation:

The project is implemented in C and consists of several functions for solving and displaying the Sudoku puzzle. The key components of the implementation are as follows:

1.Sudoku Solver Function ('solve'):

- Implements a recursive backtracking algorithm to solve the Sudoku puzzle.
- Checks for the validity of each number placement in the puzzle using the 'isSafe' function.
- Utilizes backtracking to explore different possibilities when a placement leads to an invalid solution.

2.Safety Checking Function ('isSafe'):

- Checks whether a given number placement in a particular row, column, or 3*3 grid.
- Ensures that no number is repeated in the same row, column, or subgrid

3. User Interface('main' function):

- Prompts the user to select a difficulty level (1 to 3) for the Sudoku puzzle.
- Provides predefined Sudoku puzzles based on the selected difficulty Level.
- Allows the user to input their solution for the puzzle.
- Compares the user's solution with the solved Sudoku to determine correctness.

4. Print Function ('print'):

- Prints the Sudoku puzzle to the console for the user to view.

Usage:

1. Compile the C program using a C compiler, such as GCC.
2. Run the compiled executable.
3. Follow the on-screen instructions to select the difficulty level and input your solution.

Results:

The Sudoku Solver successfully solves the Sudoku puzzle using a backtracking algorithm. The user interface allows users to interact with the program, providing a solution to the puzzle. The program then evaluates the correctness of the user's solution and provides feedback.

Conclusion:

The Sudoku Solver project, demonstrates the implementation of a backtracking algorithm to solve Sudoku puzzles. The program provides a user-friendly interface and allows users to engage with the Sudoku-solving process. The project is a practical application of algorithms and serves as an educational tool for individuals interested in Sudoku and algorithmic problem-solving.