

MINI PROJECT SYNOPSIS

Staff Use only:

Selected / Rejected	Name of the Guide	Name of the Project coordinator

1. Mini Project Topic / Title: **Sudoku Solver and Game**

Tentative:

Finalized (Guide will write):

2. Details of Group Members:

Sl no.	USN	Class	Name of Student	Mobile Number	Email Id
1	1MV19IS044	ISE 3 rd Sem	Pratham Pai BM	7022226552	prathampaibm13@gmail.com
2	1MV19IS047	ISE 3 rd Sem	Ravi Prakash Hegde	9916611795	raviappekere@gmail.com
3	1MV19IS011	ISE 3 rd Sem	Bharath GT	8296353349	bharathgt2000@gmail.com
4	1MV19IS012	ISE 3 rd Sem	Bharath TR	9480138256	trbharath7@gmail.com

3. PROJECT OVERVIEW / BACKGROUND

Introduction:

Sudoku:

- Sudoku (originally called the number-place game) is a logic-based, combinatorial number-placement puzzle. The name "Su-doku" means "Single-digit".
- A Sudoku puzzle consists of 81 cells which are divided into nine columns, rows and regions.
- The task is to place the numbers from 1 to 9 into the empty cells in such a way that in every row, column and 3×3 region each number appears only once.
- Initially a Sudoku has at least 17 given numbers but normally there are 22 to 30.
- A Sudoku is a logic-based and not a math-based puzzle. It is possible to make and solve a Sudoku puzzle with letters or even some symbols.
- A little interesting point is that there are 6,670,903,752,021,072,936,960 possible Sudoku puzzle combinations.

Project:

- Data Structures used: Linked-Stacks and 2D Arrays
- Algorithm used: Backtracking algorithm
- A C program which takes a Sudoku puzzle as an input and prints out its solution.
- And a Sudoku game application.

Need of Work / Reason for selection of this project:

Reason for choosing "Sudoku solver" as the project is -

- Interest in solving logical puzzles using C program
- To be able to simulate a logic based game as a program application
- To express our creativity and problem solving skills
- To implement and make use of various Data Structures and Algorithms to solve logical problems

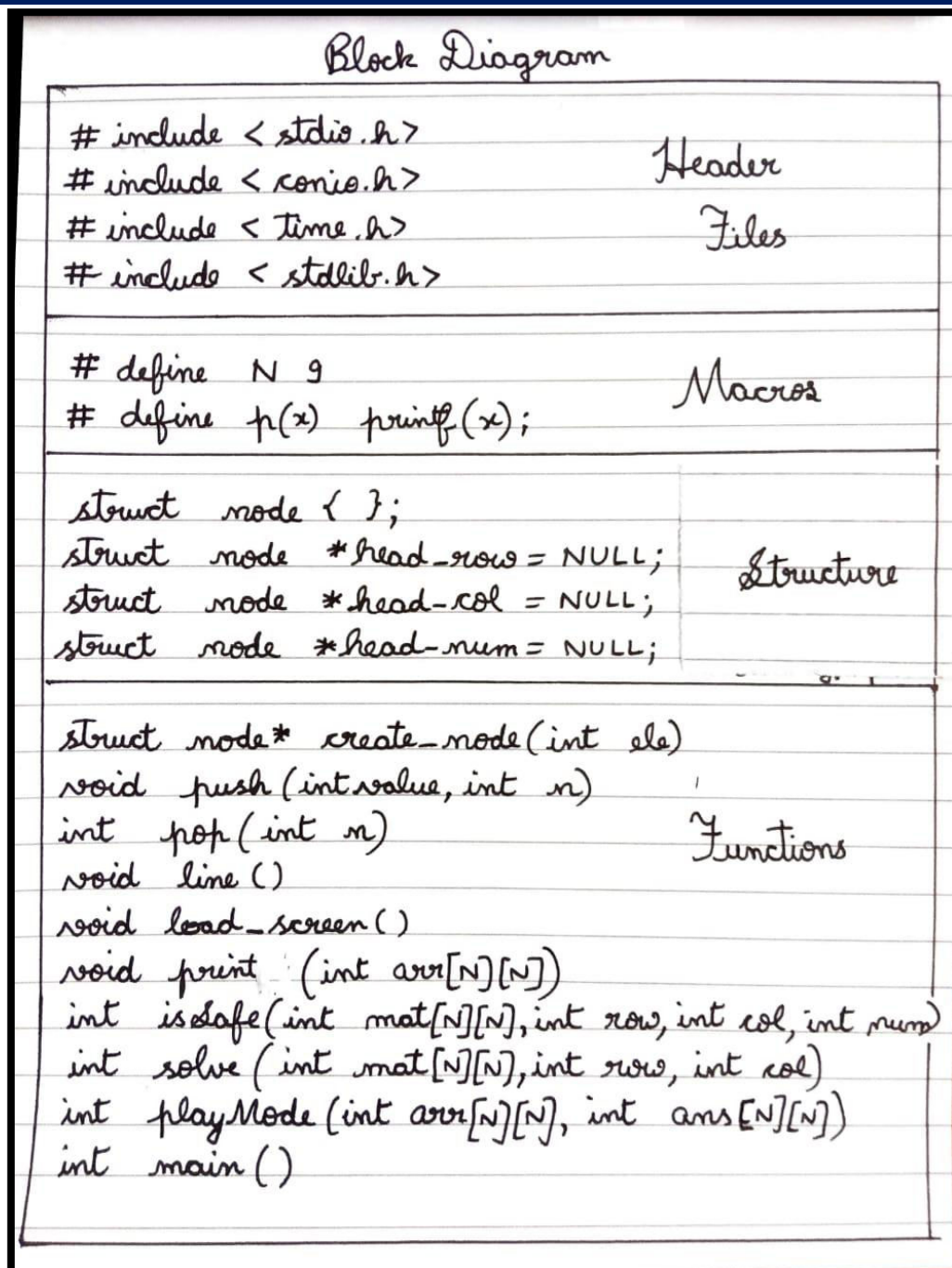
Objective:

- To be able to solve a Sudoku puzzle in a fraction of a second, and also
- To create a Sudoku game application

Problem Statement:

- To implement a C program to solve a Sudoku puzzle using appropriate data structures and algorithms
- To create a Sudoku application game using a menu driven program and functions

Block Diagram:



Explanation:

The sudoku is initially stored in a 9*9 - two dimensional matrix.

The isSafe() function accepts a row, column, number and the matrix, and checks if the number is valid in that position or not, by checking if it is already present in that row, column and the corresponding sub 3*3 matrix.

The solve() function uses the isSafe() function during the back-tracking algorithm.

The main technique used in solving the sudoku is the Back-tracking Algorithm -

- Each cell from the beginning will be tested for a number from 1-9 until it is valid
- Once a number is valid, we move onto the next cell and continue the same until the sudoku is finished.
- At some point if no number is valid (from 1-9) then we move back since our previous cell assumption was wrong, since there will be a violation.
- At some point if no number is valid for any one of the cells, for any possible combination till the end, then no solution exists for the puzzle.
- This contains various recursive functions, when there is a violation, the control returns 0 and moves back to its previous recursive function.
- Hence in short, each cell is tested for a valid number, moving "back" when there is a violation, and moving "forward" again until the puzzle is solved.

Menu driven functions to make choices based on selecting the play mode or solution mode, which is called the main menu.

Choosing solution mode, prints the solved solution (if exists) of the sudoku.

Choosing play mode, brings up a game type environment where the user can solve the sudoku himself

Choosing instructions shows the instructions for operating the application

As soon as any mode is chosen, there is a delay, which is a loading screen to bring a game type feeling. This is done using a for loop and sleep(n) function, where the control stops for n seconds (eg n=1).

The play mode is a menu driven function, using while(true) and switch statements.

The user can choose among 5 choices:

1. Input: where the user can input any number in any cell that he wishes, by entering the row, column and the number.
2. Check: the user can ask whether a particular number is correct for a given cell, and expect success or failure, which is done by comparing it with the solved answer in a separate matrix.
3. Get-answer: directly get the correct number in a particular cell by entering the corresponding row and column.
4. Undo: the user can perform undo unlimited number of times until it comes back to initial data, this is done using 3 linked-stack data structures, namely to store the row, column and number (present before giving any input), in FIFO order.
5. Exit: exit the play mode

3 linked stacks have been used for the undo operation in play mode namely for row, col, and number to store in FIFO order. So that the latest row and column that was given input, and the number that was already present in that position to be stored in the stacks.

As soon as the user exits, it prints the time taken during the entire play mode (in seconds). This is done using <time.h> header file and its inbuilt clock() function.

After exiting the solution mode or play mode, it returns back to the main menu, from where a different mode can be entered, or the application can be exit.

Benefits to the surrounding/society:

We will be able to-

- Get the solution (if exists) of any sudoku puzzle in a fraction of a second
- Check whether there exists a possible solution or not to a given sudoku puzzle
- Spend time by playing the game, learning it and also improving our logical problem solving skills

4. METHODOLOGY:

Proposed Techniques or methods to be implemented:

The sudoku is initially stored in a 9×9 - two dimensional matrix.

The technique used in solving the sudoku is done using Back-tracking Algorithm

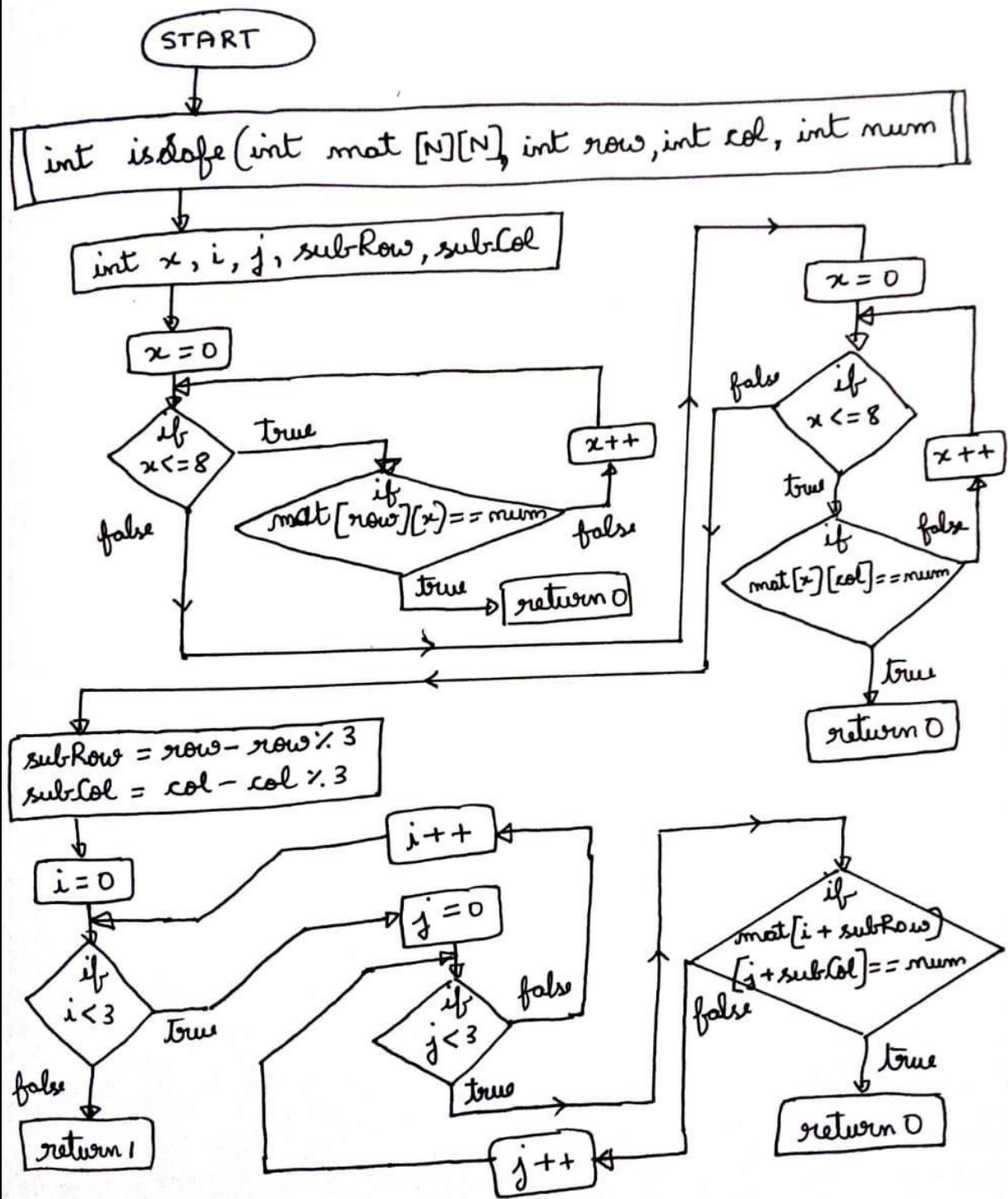
- Each cell from the beginning will be tested for a number from 1-9 until it is valid
- Once a number is valid, we move onto the next cell and continue the same until the sudoku is finished.
- At some point if no number is valid(from 1-9) then we move back since our assumption was wrong and there will be a violation, hence moving into previous recursive functions until the number is correct.
- At some point if no number is valid for any one of the cells, for any possible combination till the end, then no solution exists for the puzzle.

This contains various recursive functions, when there is a violation, the current function returns 0 and moves back to its previous recursive function.

Hence in short, each cell is tested for a valid number, moving "back" when there is a violation, and moving "forward" again until the puzzle is solved.

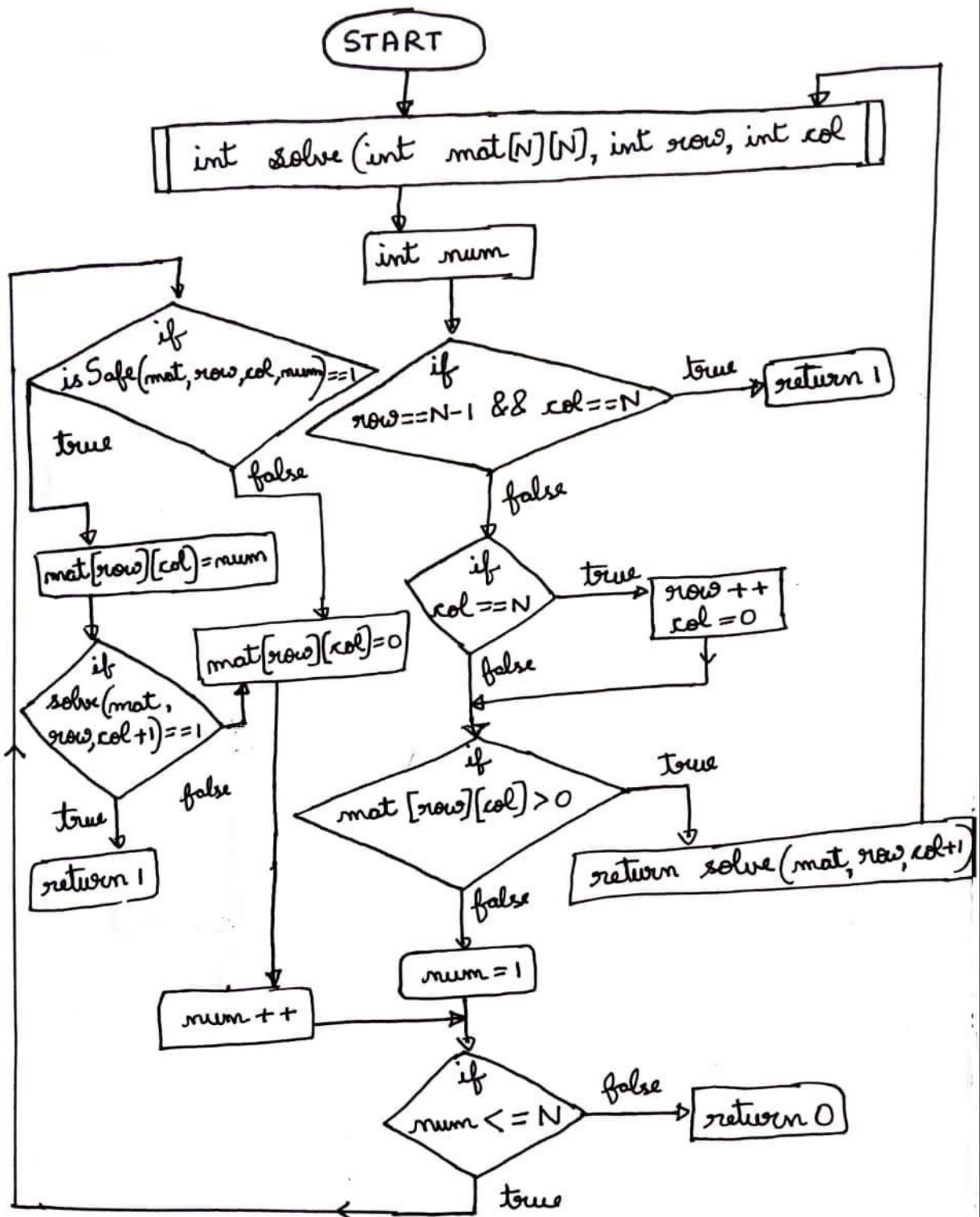
Project Flowchart(1):

FLOWCHART: for isSafe() function:



Project Flowchart(2):

FLOWCHART: for solve() function



Advantages:

- Get the solution(if exists) of any sudoku puzzle in a fraction of a second
- Check whether there exists a possible solution or not to a given sudoku puzzle
- Spend time by playing the game, learning it and also improving our logical problem solving skills, since it is a logic based game and not math based.
- Obtain the time taken for solving the game each time, and thereby trying to improve speed of solving.
- In play mode, the user can perform undo operation unlimited number of times until initial data is reached.

Limitations:

- The program cannot create random input of sudoku on its own, which means the user has to give input every time.(Since even by assigning random numbers, the solution may not exist)
- The time calculated in play mode may not be precise enough compared to the actual time.
- The clrscr() function of <conio.h> header file works only in some compilers(Eg Turbo c++). (Does not work in CodeBlocks compiler, nor in android compilers)

Applications:

- Get the solution (if exists) of any sudoku puzzle.
- Check whether there exists a possible solution or not to a given sudoku puzzle.
- Play the application as a sudoku game.
- Record amount to time user takes to solve the sudoku himself.

5. PROJECT REQUIREMENTS:

Equipments:

- Books for referring C Programming and Data Structures–
 - The C Programming Language, 2nd Edition by Brian W Kernighan, Dennis M Ritchie
 - Let Us C by Yashavant Kanetkar
 - Programming in ANSI C by E. Balagurusamy
 - Fundamentals of Data Structures in C, 2nd Edition by Horowitz, Sahni, Anderson-Freed
- An editor to create the code – Atom
- A system capable of compiling and executing C program code (Eg windows or android).
- A C compiler (Preferably Turbo C++ in windows)
- Sudoku problems for input- taken from newspapers, playstore apps and the internet.

Facilities required (include software, hardware):

- A system capable of compiling and executing C program code (Eg windows).
- C compiler preferably Turbo C++

Communication, Budget etc:

- Communication used for discussing between team members for the project
 - Whatsapp, Google Meet
- Budget – Null

6. References:

- Arnoldy, Ben. "Sudoku Strategies". The Christian Science Monitor.
- "Brain Age: Train Your Brain in Minutes a Day!". Gamerankings.com.
- Eisenhauer, William (2010). Sudoku-zilla. CreateSpace. p. 220. ISBN 978-1-4515-1049-2.
- "What is Hyper Sudoku?". www.sudoku-space.com. Retrieved August 27, 2017.
- Berthier, Denis (2007). The Hidden Logic of Sudoku. LULUPR.p.76 N. ISBN 978-1-84753-472-9.
- "NP complete – Sudoku" (PDF). Imai.is.su-tokyo.ac.jp. Retrieved October 20, 2013.
- Lewis, R. A Guide to Graph Colouring: Algorithms and Applications. Springer International Publishers, 2015.
- G. McGuire, B. Tugemann, G. Civario. "There is no 16-Clue Sudoku: Solving the Sudoku Minimum Number of Clues Problem". Arxiv.org.
- H.H. Lin, I-C. Wu. "No 16-clue Sudoku puzzles by sudoku@vtaiwan project", September, 2013.
- Royle, Gordon. "Minimum Sudoku". Retrieved February 28, 2012..
- "World Sudoku Championship 2006 Instructions Booklet" (PDF). BBC News. Retrieved May 24, 2010.
- "Thomas Snyder wins World Sudoku Championship". US Puzzle Team. March 31, 2007. Retrieved April 18, 2008.
- "WSPC 2017 – Logic Masters India". wspc2017.logicmastersindia.com.
- "World Sudoku Championships | WPF". www.worldpuzzle.org.

Signature of Mini Project Co-ordinator:

Signature of H.O.D.: