

“Sudoku Game”

(OS Lab Project Report)

Submitted By:

Muhammad Usama Shaheer 2019-CS-605

Daud Ahmad 2019-CS-620

Department of Computer Science,
University of Engineering and Technology, Lahore
(New Campus)

Table of Contents

Introduction	3
History:	3
Problem:	3
Solution:	4
Related Study	5
Methodology	5
Description:	5
Block Diagram:	8
Result	9
Starting the Game:	9
MakeFile:	10
First Menu:	10
Demo:	11
Play the Game:	12
Submit your Puzzle:	13
Wrong Submitted Puzzle:	14
Edit Your Puzzle:	16
Future Work	18
References	19

Introduction

Our project is Sudoku game. It has 9 by 9 grid having 1 to 9 digits with no repetition and 3 by 3 sub grid also having 1 to 9 digits with no repetition. we are basically going to implement a solution for the validation of the Sudoku Puzzle using treads on Linux Shell in C language.

History:

The History of Sudoku is quite interesting and itself a Puzzle. The name of the Sudoku comes from Japan. It is comprising of two Japanese words **Su** (meaning 'number') and **duko** (meaning 'single'). Interesting thing here is that it was not invented in Japan. It originated from Switzerland and come to the Japan by the way of America. Sudoku has been come from many different ancient puzzles including Magic Squares, Chinese Puzzle etc. In the past, people have had a great interest in solving this puzzle and also it has been part of several newspapers. Initially it was played on some card board or on some paper. But now in this era, we can also play it on digital devices like computer and mobile applications as well.

There are some of the reasons that urges us to play this puzzle:

- Sudoku puzzle offers a relaxing way to take a break from your world. It refreshes us and allows us to meet the other commitments with renewed energy and vigor of our daily life.
- The American Alzheimer's Association has recommended Sudoku Puzzle as a "Brain game" that might help reduce the risk of Alzheimer's disease. Some of the Scientists also urged that playing puzzle games like Sudoku may reduce the risk of dementia (the disease of memory loss).
- Sudoku puzzle is such a type of game that can be played at any time. Moreover, it's not a kind of game to which people become addictive rather playing this game make people more intelligent.
- Playing Sudoku Puzzle may improve your problem-solving skills and helps you finding the solution of your daily life problems easily.

Problem:

In Sudoku puzzle, we use a 9×9 grid in which each column and row, as well as each of the nine 3×3 sub-grids (also called "Houses"), must contain all of the digits 1 through 9. It cannot contain a number greater than 9 or smaller than 1. So that each row sum; column sum; and 3×3 grid sum become equal to 45. The main problem here in this game is to take users input, interpret it whether the provided input is a valid solution of the Sudoku Puzzle or not.

Solution:

We are using Operating System Programming's technique Multi-threading for the Solution of the Puzzle. The reason behind using this technique is the maximum utilization of CPU and decreasing the response time. The Actual functionality providing functions in our project are three that are called by 27 threads. 9 threads calling one function i.e., 9 threads are calling a function that validates all the rows, 9 threads are calling a function that validates all columns and the remaining 9 threads are calling a function to validate 9 sub 3 x 3 matrices. **Figure 1** presents an example of a valid Sudoku Puzzle. We have also tried to provide a good and interactive interface to the player so that his interest in playing the game may be preserved.

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

Figure 1: Example of the Valid Sudoku Puzzle

Related Study

We have noticed that there is large number of People that had worked on this Project. Even some of the People had made a complete Graphical user interfaced Project of Sudoku Game. *Carlos Perea* [1] had solved this puzzle using a similar technique in which he is using 11 threads to solve the puzzle. But in this way, the method has become more complex. I think working with more threads can be more efficient so I've used 27 threads. Moreover, the result generated from each thread can be used for further clarification of result presented to the player.

Brandon Bremner [2] had solved this problem in the same way mentioned above as done by the *Carlos Perea*. But the thing different he has done is that he reading input from a text file. User only have to put his solved puzzle in a text file and then the file will be checked by the program.

Methodology

Description:

All the path through which we have made our project is presented in the Block Diagram (the next heading). These steps are also described in the following points:

- When we thought about OS project then there is a large number of Projects are available there can be created. So, selecting a Correct Project and fulfilling its needs is a major part.
- Then we prepared a Proposal of the Project selected "Sudoku Puzzle". Our Project was selecting in the first go, only a minor change was assigned to us to add in our project.
- Now there was a time to understand the problem clearly. Understanding the problem is the half solution of that problem.
- We decided to solve the problem using multi-threading technique. For that we used 27 threads that works in parallel for solving the puzzle.
- We Created 27 threads in our project i.e., 9 threads were working for the validity of 9 Rows, 9 threads were working for the validity of 9 Columns, and the remaining of 9 threads was working for the validity of 3 x 3 matrix that the values in that area is according to the given criteria or not:
 - Numbers entered are in between 1 through 9.
 - Any number should be not repeated. All the numbers should appear once in its section.
 - It should not contain any character, symbol or number smaller than 1 or greater than 9.
- For the Validity of Rows, we've created a function that check for all the rules mentioned above on all the entries of that particular row for which it is called,

then saved its result as **0** or **1** in an array of size 27. Similarly, one function was created for the validity of Columns, and one function was created for the validity of 3 x 3 matrix and saving its result in the array. So that we can elaborate the result to the player and he can better play next time.

- Main Logic for the Validity of rows and Column is a bit similar:

```
indexes *params = (indexes *) param;
int r = params->row;
int c = params->column;

int checkArray[9] = {0};
for(int i = 0; i < 9; i++)
{
    int value = sudokuBoard[r][i];
    if(value < 1 || value > 9 || checkArray[value - 1] == 1)
    {
        pthread_exit(NULL);
    }
    else
    {
        //Maintaining record for each value occurred only once
        in the whole Row
        checkArray[value - 1] = 1;
    }
}
//Maintaining record for each Row in the isValid array
isValid[9 + r] = 1;
```

- While Logic for the Validity of 3 x 3 matrix is interesting one:

```
indexes *params = (indexes *) param;
int r = params->row;
int c = params->column;

int checkArray[9] = {0};
for(int i = r; i < r + 3; i++)
{
    for(int j = c; j < c + 3; j++)
    {
        int value = sudokuBoard[i][j];
        if(value < 1 || value > 9 || checkArray[value - 1] ==
1)
        {
            pthread_exit(NULL);
        }
        else
        {
```

```

//Maintaining record for each value occurred only
once in the whole Row
    checkArray[value - 1] = 1;
}
}
}
//Maintaining record for each 3 x 3 matrix in the isValid array
isValid[r + c/3] = 1;

```

- One requirement assigned from Prof. was that from Terminal our Game should not accept any string, negative value, symbol or a 2 digit value. Also it should go to the next line immediately when inputs for the current line ends. Code for this problem was also a challenge to implement:

```

printf("\nEnter values for Sudoku Puzzle row by row");
for(int i = 0; i < 9; i++)
{
    printf("\nEnter for Row %d: ", i+1);
    for(int j = 0; j < 9; j++)
    {
        temp = getche();
        printf(" ");
        value[0] = temp;
        value[1] = '\0';
        //printf("atoi
value%d", atoi(value));
        if(atoi(value) < 1 || atoi(value)
> 9)
        {
            printf("\nInvalid          value
\'%c\' \nRe-enter...!", temp);
            i--;
            break;
        }
        sudokuBoard[i][j] = (int)temp - 48;

        //scanf("%d", &sudokuBoard[i][j]);
    }
}
printf("\n\n\tValues          entered
successfully...\n");

```

- After the main logic is implemented, we've to provide a good interface to the game so that the game may be interactive and that the players interest may also be maintained.

- Alongside this we've also provided a Demo to the players coming first time for playing the game. In that demo, we have presented the rules for playing the game also a solved Puzzle is provided for the better understanding of the game.
- Finally, we checked out project on several inputs that it is working according to the requirement or not and fixed the problems encountered and then presented.

Block Diagram:

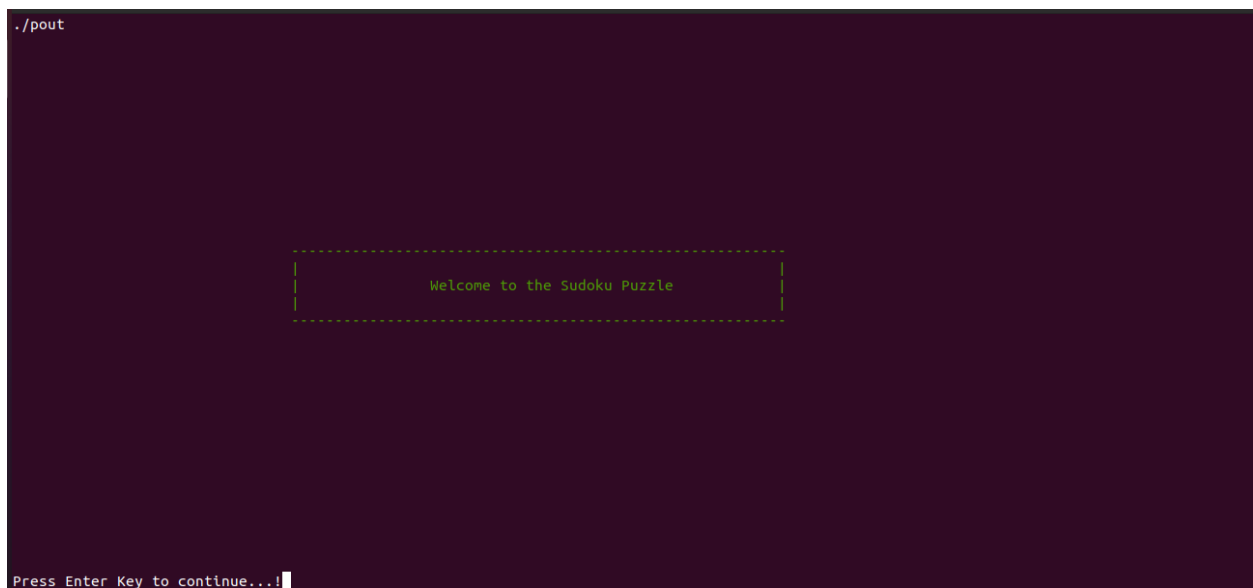


Result

Now, we'll present our Project through snapshots of the output screen.

Starting the Game:

If one opens the game directory in terminal, he needs to open the game just by typing 'make'. Against this MakeFile of the program is executed. After which a Welcome screen appears.

A terminal window with a dark purple background. In the top-left corner, the text `./pout` is visible. In the center, the text `Welcome to the Sudoku Puzzle` is displayed in a light green monospace font, enclosed within a dashed green rectangular border. At the bottom-left, the text `Press Enter Key to continue...!` is shown in white, followed by a white cursor icon.

```
./pout

Welcome to the Sudoku Puzzle

Press Enter Key to continue...!
```

It needs enter key to pressed to go to the next screen.

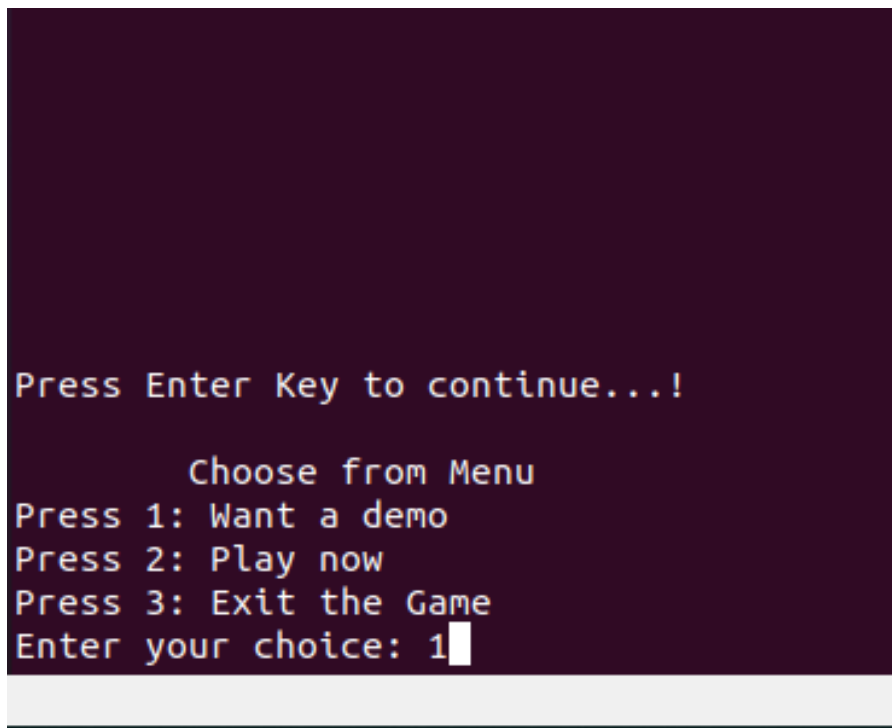
MakeFile:

MakeFile contains all the commands to Compile the project and run it and after the Game ends it immediately delete the output file. So, a new output file is created every time a player tries to play the game.

```
1 all: compile exec clean
2 compile:
3     gcc main.c utilities.c -pthread -o pout
4 exec:
5     ./pout
6 clean:
7     rm -rfv pout
```

First Menu:

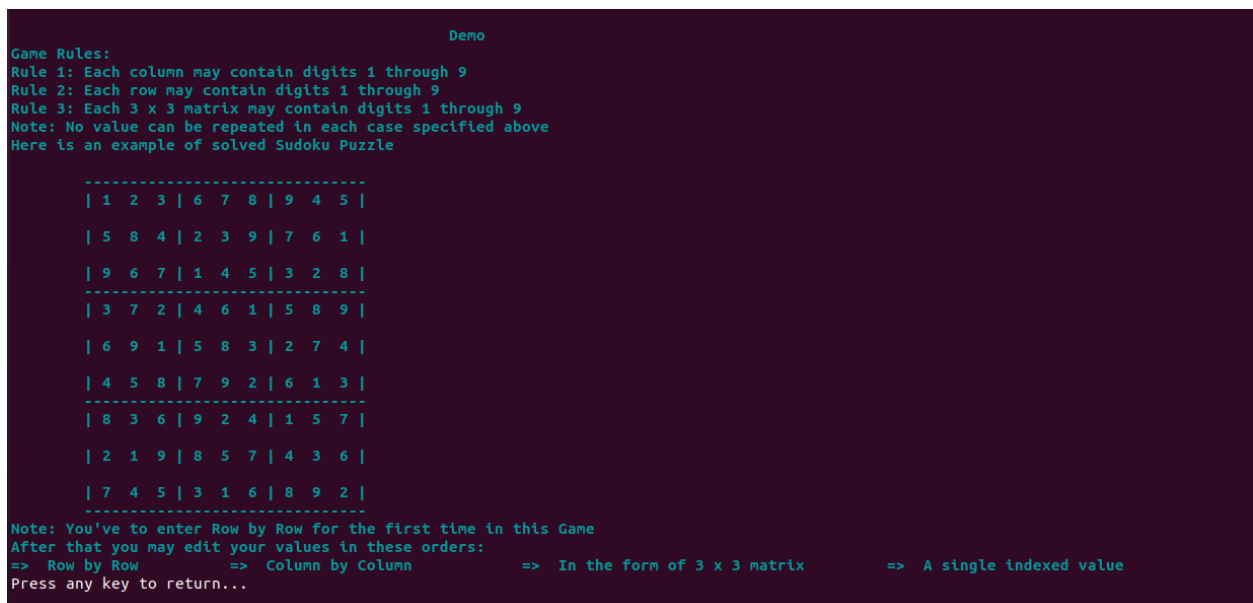
The First Menu the player face is as follows:



Here we've chosen 1 for a Demo.

Demo:

In the Demo screen we've presented a brief demo to the players who has first interaction with the game. All the rules related to the game are mention here. Also, a solved Puzzle is presented so that the player may better understand the game and enjoy.



A key stroke is waiting for the user to go to the Next screen.

Play the Game:

When any key is pressed then the above menu is again presented to the player. Now the Second option here is Play Now. Here the Player can input the values for the Puzzle. Also, a time interval of 5 Mins. is given to the Player in which he can change and manipulate his entries in the Puzzle. After the specified time the Game will not show the result to the player, when he'll submit the game.

```
Press any key to return...
```

```
        Choose from Menu
```

```
Press 1: Want a demo
```

```
Press 2: Play now
```

```
Press 3: Exit the Game
```

```
Enter your choice: 2
```

```
Your Time has started Now, you have total of 5 mins.
```

```
Enter values for Sudoku Puzzle row by row
```

```
Enter for Row 1: 1 2 4 5 3 9 1 8 7
```

```
Enter for Row 2: 5 1 9 7 2 8 6 3 4
```

```
Enter for Row 3: 8 3 7 6 1 4 2 9 5
```

```
Enter for Row 4: 1 4 3 8 6 5 7 2 9
```

```
Enter for Row 5: 9 5 8 2 5 7 3 6 1
```

```
Enter for Row 6: 7 6 2 3 9 1 4 5 8
```

```
Enter for Row 7: 3 7 1 9 5 6 8 4 2
```

```
Enter for Row 8: 4 9 6 1 8 2 5 7 3
```

```
Enter for Row 9: 2 8 5 4 7 3 9 1 6
```

```
        Values entered successfully...
```

```
        Choose from Menu
```

```
Press 1: Submite your puzzle
```

```
Press 2: View your Sudoku Puzzle values
```

```
Press 3: Edit your puzzle
```

```
Enter your choice: █
```

Submit your Puzzle:

If the player has submitted the puzzle in the given time interval then his puzzle will be accepted and interpreted. Otherwise, it will not be interpreted, rejected as out of time as presented in the next picture though the solution is correct but the time elapsed so the Game has not shown output. Rather it's just displays "You are out of time."

```
Your Time has started Now, you have total of 5 mins.

Enter values for Sudoku Puzzle row by row
Enter for Row 1: 6 2 4 5 3 9 1 8 7
Enter for Row 2: 5 1 9 7 2 8 6 3 4
Enter for Row 3: 8 3 7 6 1 4 2 9 5
Enter for Row 4: 1 4 3 8 6 5 7 2 9
Enter for Row 5: 9 5 8 2 4 7 3 6 1
Enter for Row 6: 7 6 2 3 9 1 4 5 8
Enter for Row 7: 3 7 1 9 5 6 8 4 2
Enter for Row 8: 4 9 6 1 8 2 5 7 3
Enter for Row 9: 2 8 5 4 7 3 9 1 6

    Values entered successfully...

    Choose from Menu
Press 1: Submit your puzzle
Press 2: View your Sudoku Puzzle values
Press 3: Edit your puzzle
Enter your choice: 1 Testing values and counting time...
You have taken 0Mins. 4 Seconds to Solve the Puzzle

Your Submitted Solution is Absolutly Correct...

Press any Key to close the Application...!
█
```

Late Submitted Result:

Your Time has started Now, you have total of 5 mins.

Enter values for Sudoku Puzzle row by row

Enter for Row 1: 1 2 4 5 3 9 1 8 7

Enter for Row 2: 5 1 9 7 2 8 6 3 4

Enter for Row 3: 8 3 7 6 1 4 2 9 5

Enter for Row 4: 1 4 3 8 6 5 7 2 9

Enter for Row 5: 9 5 8 2 5 7 3 6 1

Enter for Row 6: 7 6 2 3 9 1 4 5 8

Enter for Row 7: 3 7 1 9 5 6 8 4 2

Enter for Row 8: 4 9 6 1 8 2 5 7 3

Enter for Row 9: 2 8 5 4 7 3 9 1 6

Values entered successfully...

Choose from Menu

Press 1: Submite your puzzle

Press 2: View your Sudoku Puzzle values

Press 3: Edit your puzzle

Enter your choice: 1 Testing values and counting time...

You're out of time

Try Better next Time...

Press any Key to close the Application...!



Wrong Submitted Puzzle:

In this example, the submitted puzzle is wrong. The wrong values placement 3 x 3 matrices are given in the picture. For all the cases, it submits the puzzle and presents the result, our game also tells the user that how much time he has taken to solve the puzzle. Also, his used output file is deleted at the end of the Game as shown in the following picture.

Your Time has started Now, you have total of 5 mins.

Enter values for Sudoku Puzzle row by row

Enter for Row 1: 5 3 4 6 7 8 9 1 2

Enter for Row 2: 6 7 2 1 9 5 3 4 8

Enter for Row 3: 1 9 8 3 4 2 5 6 7

Enter for Row 4: 8 5 9 7 6 1 4 2 3

Enter for Row 5: 4 2 6 8 5 3 7 9 1

Enter for Row 6: 9 6 1 5 3 7 2 8 4

Enter for Row 7: 2 8 7 4 1 9 6 3 5

Enter for Row 8: 3 4 5 2 8 6 1 7 9

Enter for Row 9: 7 1 3 9 2 4 8 5 6

Values entered successfully...

Choose from Menu

Press 1: Submite your puzzle

Press 2: View your Sudoku Puzzle values

Press 3: Edit your puzzle

Enter your choice: 1 Testing values and counting time...

You have taken 0Mins. 4 Seconds to Solve the Puzzle

Your Submitted Solution is not Correct...

Errors are in:

4 Matrix, 5 Matrix, 6 Matrix, 7 Matrix, 8 Matrix, 9 Matrix,

Press any Key to close the Application...!

rm -rfv pout

removed 'pout'

daud@daud-VirtualBox:~/Desktop/project/Edited\$

Edit Your Puzzle:

On Selecting 3 in the second Menu, you can Edit your Puzzle values in different ways.

- In Row order
- In Column Order
- In Matrix Order
- A Single Indexed Value

For Row, Column and 3 x 3 matrix editing you have to provide its index/number in the Puzzle Board i.e., in this case we're editing our values for Row 4.

```
Your Time has started Now, you have total of 5 mins.
```

```
Enter values for Sudoku Puzzle row by row
```

```
Enter for Row 1: 6 2 4 5 3 9 1 8 7
```

```
Enter for Row 2: 5 1 9 7 2 8 6 3 4
```

```
Enter for Row 3: 8 3 7 6 1 4 2 9 5
```

```
Enter for Row 4: 1 4 3 8 6 5 7 2 9
```

```
Enter for Row 5: 9 5 8 2 4 7 3 6 1
```

```
Enter for Row 6: 7 6 2 3 9 1 4 5 8
```

```
Enter for Row 7: 3 7 1 9 5 6 8 4 2
```

```
Enter for Row 8: 4 9 6 1 8 2 5 7 3
```

```
Enter for Row 9: 2 8 5 4 7 3 9 1 6
```

```
Values entered successfully...
```

```
Choose from Menu
```

```
Press 1: Submite your puzzle
```

```
Press 2: View your Sudoku Puzzle values
```

```
Press 3: Edit your puzzle
```

```
Enter your choice: 3
```

```
Choose from Menu
```

```
Press 1: Edit Row
```

```
Press 2: Edit Column
```

```
Press 3: Edit Matrix
```

```
Press 4: Edit single value
```

```
Press 5: View your Sudoku Puzzle values
```

```
Press 6: Submite your puzzle
```

```
Enter your choice: 1
```

```
Enter Row Number: 4
```

```
Enter Values: 1 4 3 8 6 5 7 2 9
```

```
Values Entered press any key to return...
```



But in case when the player wants to edit a single value, then he has to provide row number and column number in the Puzzle Board.

```
Your Time has started Now, you have total of 5 mins.
```

```
Enter values for Sudoku Puzzle row by row
```

```
Enter for Row 1: 6 2 4 5 3 9 1 8 7
```

```
Enter for Row 2: 5 1 9 7 2 8 6 3 4
```

```
Enter for Row 3: 8 3 7 6 1 4 2 9 5
```

```
Enter for Row 4: 1 4 3 8 6 5 7 2 9
```

```
Enter for Row 5: 9 5 8 2 4 7 3 6 1
```

```
Enter for Row 6: 7 6 2 3 9 1 4 5 8
```

```
Enter for Row 7: 3 7 1 9 5 6 8 4 2
```

```
Enter for Row 8: 4 9 6 1 8 2 5 7 3
```

```
Enter for Row 9: 2 8 5 4 7 3 9 1 6
```

```
Values entered successfully...
```

```
Choose from Menu
```

```
Press 1: Submite your puzzle
```

```
Press 2: View your Sudoku Puzzle values
```

```
Press 3: Edit your puzzle
```

```
Enter your choice: 3
```

```
Choose from Menu
```

```
Press 1: Edit Row
```

```
Press 2: Edit Column
```

```
Press 3: Edit Matrix
```

```
Press 4: Edit single value
```

```
Press 5: View your Sudoku Puzzle values
```

```
Press 6: Submite your puzzle
```

```
Enter your choice: 1
```

```
Enter Row Number: 4
```

```
Enter Values: 1 4 3 8 6 5 7 2 9
```

```
Values Entered press any key to return...
```

```
█
```

View Your Puzzle:

Here by choosing 5, player can view his puzzle in a nice and decent way. All the entries of the player are presented in a uniform and well furnished way.

```

    Choose from Menu
Press 1: Edit Row
Press 2: Edit Column
Press 3: Edit Matrix
Press 4: Edit single value
Press 5: View your Sudoku Puzzle values
Press 6: Submite your puzzle
Enter your choice: 5

    Sudoku Board

-----
| 6  2  4 | 5  3  9 | 1  8  7 |
| 5  1  9 | 7  2  8 | 6  3  4 |
| 8  3  7 | 6  1  4 | 2  9  5 |
-----
| 1  4  3 | 8  6  5 | 7  2  9 |
| 9  5  8 | 2  4  7 | 3  6  1 |
| 7  6  2 | 3  9  1 | 4  5  8 |
-----
| 3  7  1 | 9  5  6 | 8  4  2 |
| 4  9  6 | 1  8  2 | 5  7  3 |
| 2  8  5 | 4  7  3 | 9  1  6 |
-----

```

Future Work

If you ask us that our project may extended? Then answer is YES. One can add many features to this project if he wants. This is basically a validator that only check for the valid value of the player (according to the rules of the Sudoku game).

In future, the feature that can be implemented may be that the user can input it entries in .text file instead of input on terminal if he only wants to check its entries for validity. This string then converts in to integer and program check all entries whether they are valid or not by using validity functions.

In future this can also be implanted that on some of the positions we already specify correct values. Player needs to follow those entries and full fill the Puzzle using these entries. After the user inputs all the entries in the puzzle the game then validate the Puzzle and present the result to the Payer.

References

[1] <https://github.com/cfperea/multithreaded-sudoku>

[2] <https://github.com/BruceMacD/Sudoku-Checker>
