SUDOKU SOLVER

by

SAADHIKHA SHREE S	19BPS1075
RAJASHRI MAHATO	19BPS1130

A project report submitted to

Dr. FLORENCE GNANA POOVATHY J.

SCHOOL OF ELECTRONICS ENGINEERING

in partial fulfilment of the requirements for the course of

CSE2006 – MICROPROCESSOR AND INTERFACING

in

B. Tech. COMPUTER SCIENCE AND ENGINEERING spl. in CYBER PHYSICAL SYSTEM

Vandalur – Kelambakkam Road

Chennai – 600127

NOVEMBER 2020

BONAFIDE CERTIFICATE

Certified that this project report entitled “**SUDOKU SOLVER”** is a bonafide work of **SAADHIKHA SHREE S – 19BPS1075 and RAJASHRI MAHATO -19BPS1130** who carried out the Project work under my supervision and guidance for **CSE2006- MICROPROCESSOR AND INTERFACING.**

Dr. FLORENCE GNANA POOVATHY J.

Assistant Professor Sr.

School of Electronics Engineering (SENSE),

VIT University, Chennai

Chennai – 600 127.

ABSTRACT

The Sudoku solver is an emu 8086 controlled program used to solve 3x3 Sudoku puzzles. A typical, standard Sudoku puzzle contains 81 cells, in a 9×9 grid, and has 9 boxes, each box being the intersection of the first, middle, or last 3 rows, and the first, middle, or last 3 columns. Each cell may contain a number from one to nine, and each number can only occur once in each row, column, and box. A Sudoku starts with some cells containing numbers (clues), and the goal is to solve the remaining cells. Proper Sudokus have one solution. It uses simple techniques and algorithms of artificial intelligence (AI) and machine learning. Sudoku solver uses a step-by-step mechanism to solve a 3x3 Sudoku puzzle to yield correct results all the time. It uses the basic principle of solving any Sudoku puzzle i.e. Recursion. It solves the certain cells first and foremost and leaves the uncertain cells unfilled. It repeats the process of filling the certain cells and leaving the uncertain ones until all the cells are filled. This mechanism ensures that the Sudoku solver is always 100 percent efficient in solving any type of sudoku puzzles given. Sudoku is a logic-based, combinatorial number-placement puzzle. In classic sudoku, the objective is to fill a 9×9 grid with digits so that each column, each row, and each of the nine 3×3 sub grids that compose the grid (also called "boxes", "blocks", or "regions") contain all of the digits from 1 to 9. The puzzle setter provides a partially completed grid, which for a well-posed puzzle has a single solution.

ACKNOWLEDGEMENT

We wish to express our sincere thanks and deep sense of gratitude to our project guide, **Dr. FLORENCE GNANA POOVATHY J.,** Assistant Professor Sr., School of Electronics Engineering, for her consistent encouragement and valuable guidance offered to us in a pleasant manner throughout the course of the project work.

We are extremely grateful to **Dr. SIVASUBRAMANIAN. A,** Dean of School of Electronics Engineering, VIT Chennai, for extending the facilities of the School towards our project and for his unstinting support.

We express our thanks to our Head of the Department **Dr. JAGADEESH KANNAN** for his support throughout the course of this project.

We also take this opportunity to thank all the faculty of the School for their support and their wisdom imparted to us throughout the course.

We thank our parents, family, and friends for bearing with us throughout the course of our project and for the opportunity they provided us in undergoing this course in such a prestigious institution.

S. SaadhikhaShree   Rajashri Mahato.

SAADHIKHA SHREE S   RAJASHRI MAHATO

NAME WITH SIGNATURE   **NAME WITH SIGNATURE**

TABLE OF CONTENTS

1. INTRODUCTION

1.1 OBJECTIVES AND GOALS

OBJECTIVES :

(a) Each row should have unique numbers from 1–9 or empty spaces.
(b) Each column should have unique numbers from 1–9 or empty spaces.
(c) Each sub-grid of 1–9 should have the numbers 1–9 or empty spaces.

GOALS :

(a) Fill in the numbers from 1–9 exactly once in each row, column, and 3x3 region.
(b) The board is already filled, meaning there is no white space.
(c) There are no more empty spots left for the algorithm to check, then the current candidate does not reach our goal.

1.2 SCOPE

The scope of the proposed Project is to increase the Thinking Capability.
The Game having all the records which we perform, in playing, we can Select, Easy, Moderate and Hard level according to our choice. We can make our own Sudoku and at any Step we can go back to One Step as well as we can see the Solution of it. It is manually a very difficult job to perform and it needs a lot of recalling, reminding and mathematical calculation. The game of "Sudoku" helps to increase mental thinking, vision etc.

1.3 TOOLS USED

(a) The Microsoft Assembler (commonly known as MASM) is an industrial software development tool that has been maintained and updated for over 30 years by a major operating system vendor. It has never been softened or compromised into a consumer friendly tool and is designed to be used by professional programmers for operating system level code and high performance object modules, executable files and dynamic link libraries.

(b) Intel 8086 microprocessor is the enhanced version of Intel 8085 microprocessor. It was designed by Intel in 1976.
The 8086 microprocessor is a16-bit, N-channel, HMOS microprocessor. Where the HMOS is used for "High-speed Metal Oxide Semiconductor".

(c) DOSBox is a program emulator which emulates an IBM PC compatible computer running a DOS operating system. DOSBox emulates a full x86 pc with sound and DOS.

1.4 FEATURES

The traditional long method for reaching a solution would be to assign every space in the same row, column, and box of the given value, every possible solution for each undefined square. After analyzing every space, possible solutions will be eliminated because new information will contradict old information and create a more refined list of possibilities. The issue with this method is the fact that the amount of information present from the start of this solution method is much more than the information present at the start of the elimination method and therefore is a less practical method of solution. Hence we use our method of working.

Considering several rules for this game and comparing the time complexities of the same in order to come up with the most efficient rule to solve the sudoku puzzle.

The rules considered are :

(a) ONLY CHOICE RULE: In the simplest case you have a group that has eight squares allocated leaving only one remaining choice available; so the remaining number must go in that empty square. From then on that particular number is filled first wherever possible and the next number is chosen.

(b) ONLY SQUARE RULE: Only Square Rule is a straight forward rule. Here if a group has seven squares allocated with only two numbers left to allocate it is often the case that an intersecting (or shared) group forces a number to go in one of the squares and not the other one. You are left with an “only square” within a group for a number to go in.

(c) SINGLE POSSIBILITY RULE: In cases when you look at individual squares you will often find that there is only one possibility left for the square. Such squares are identified and filled first. Finding out square by square in such case would need more traversing through the given sudoku board. But if such lone positions are filled in the initial stages it might help us for reaching the goal sooner. This method is termed as Single possibility rule.

(d) BACKTRACKING: This is proved to be the simplest and fetches the most elegant solution. Backtracking is a general algorithm for finding all (or some) solutions to a problem that incrementally builds candidates to the solution. As soon as it determines that a candidate can not possibly be the solution to the problem, it abandons it.

2. DESIGN

2.1 DIAGRAM

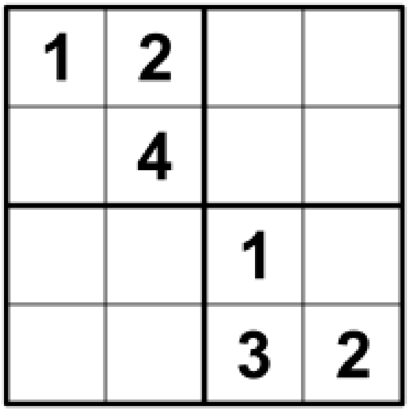


FIG. 1: 4x4 sudoku puzzle

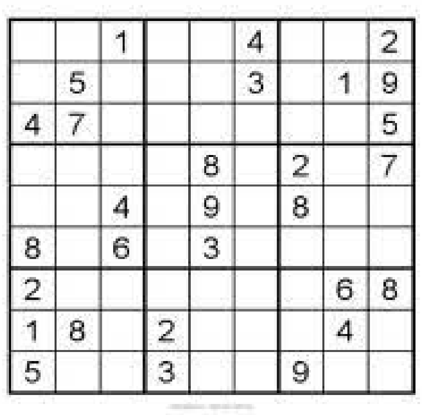


FIG. 2 : 9x9 sudoku puzzle

2.2 PAPER PEN APPROACH

Figures 3- 10 represent the method of solving a 3x3 subgrid in a sequential pattern .

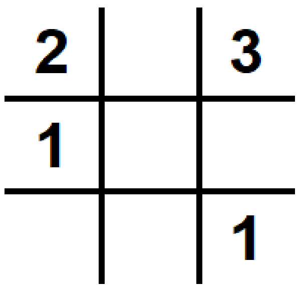


FIG. 3

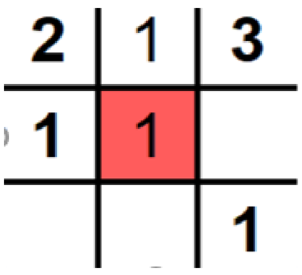


FIG. 4

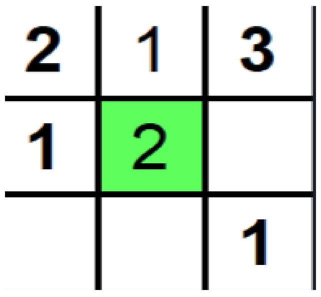


FIG. 5

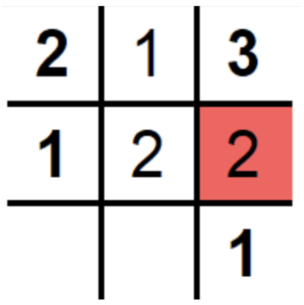


FIG. 6

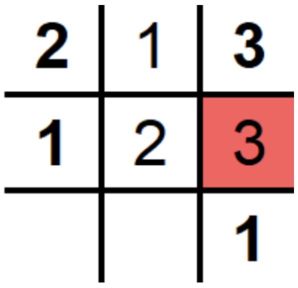


FIG. 7

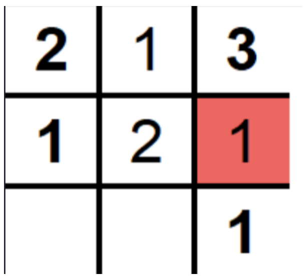


FIG. 8

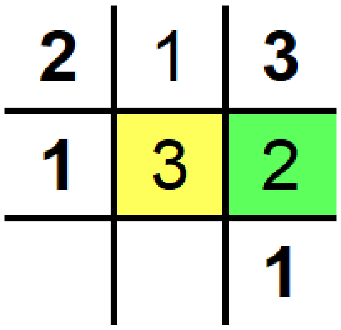


FIG. 9

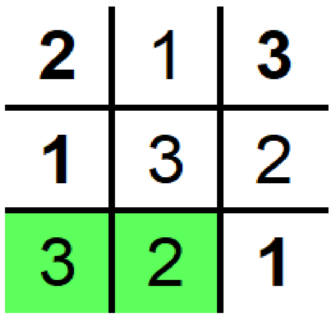


FIG. 10

We start off by listing all the empty spots. If we label each cell in the grid with a pair of numbers (x,y) and mark the first cell (1,1). Now let us consider the same grid with one small change. Replacing the 1 in cell (3,3) with a 2 renders the grid unsolvable. Similarly, removing hints from cells (2,1) and (3,3) allows for multiple solutions. But since this algorithm has a single goal, it stops after the first solution is reached.

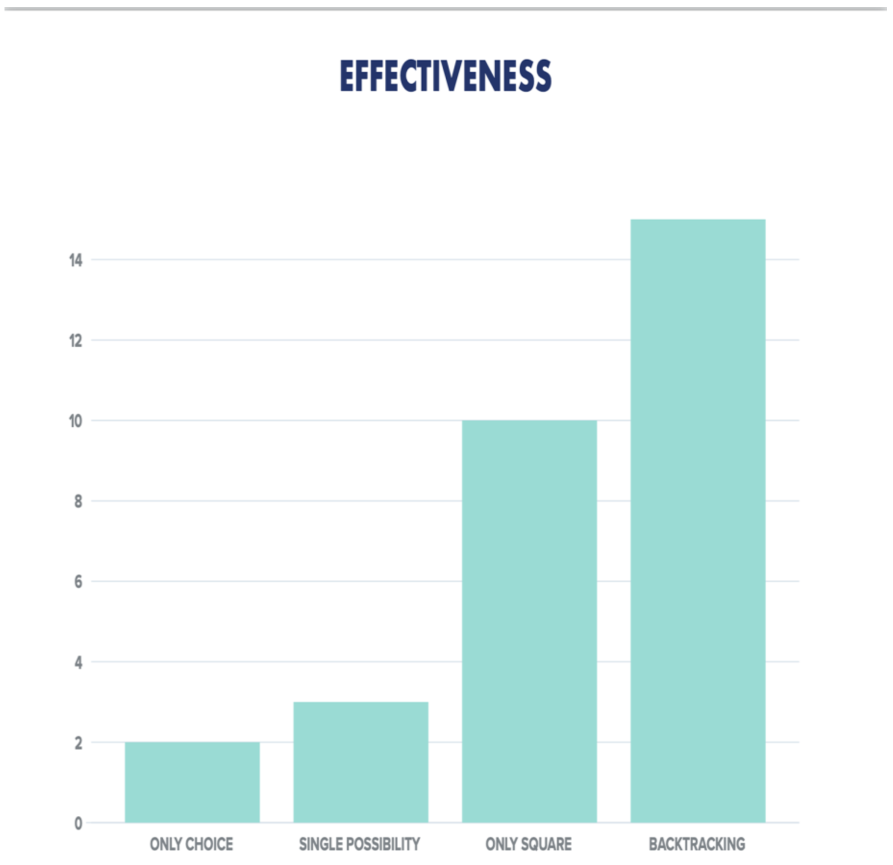
3. SOFTWARE

3.1 SOFTWARE ANALYSIS

PARAMETERS :

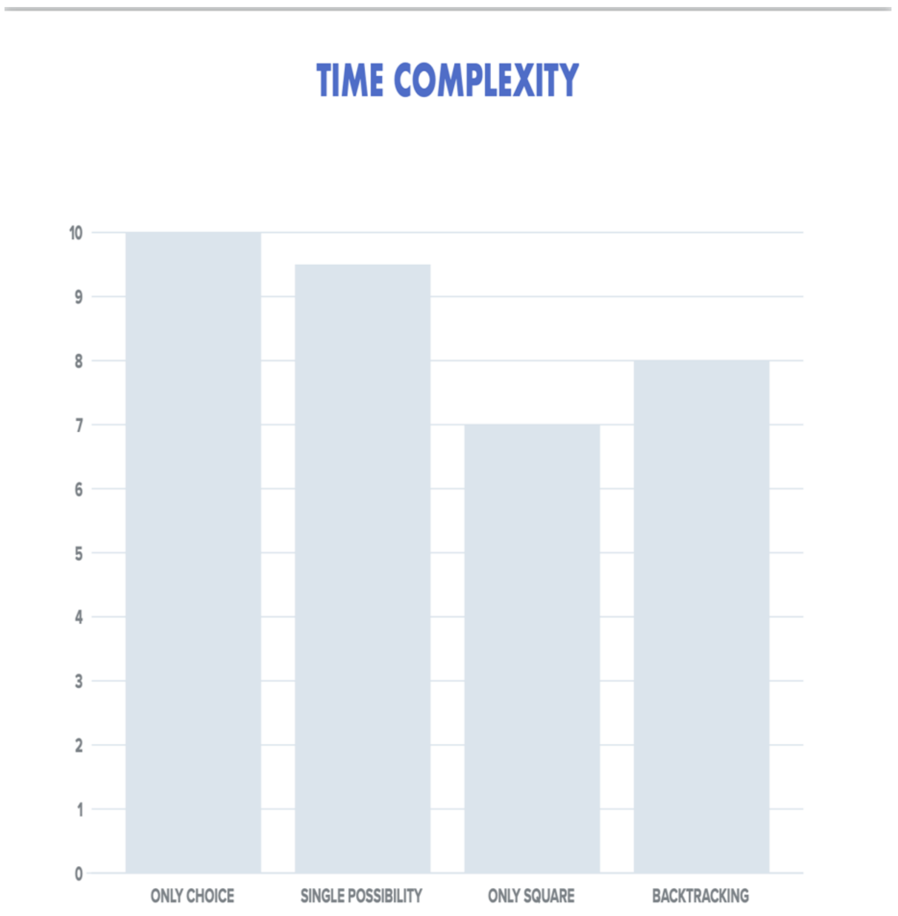
We need to standardize some basic parameters to compare and come up with the best solution possible to solve our problem. Hence we have considered the following parameters:

(a) EFFECTIVENESS



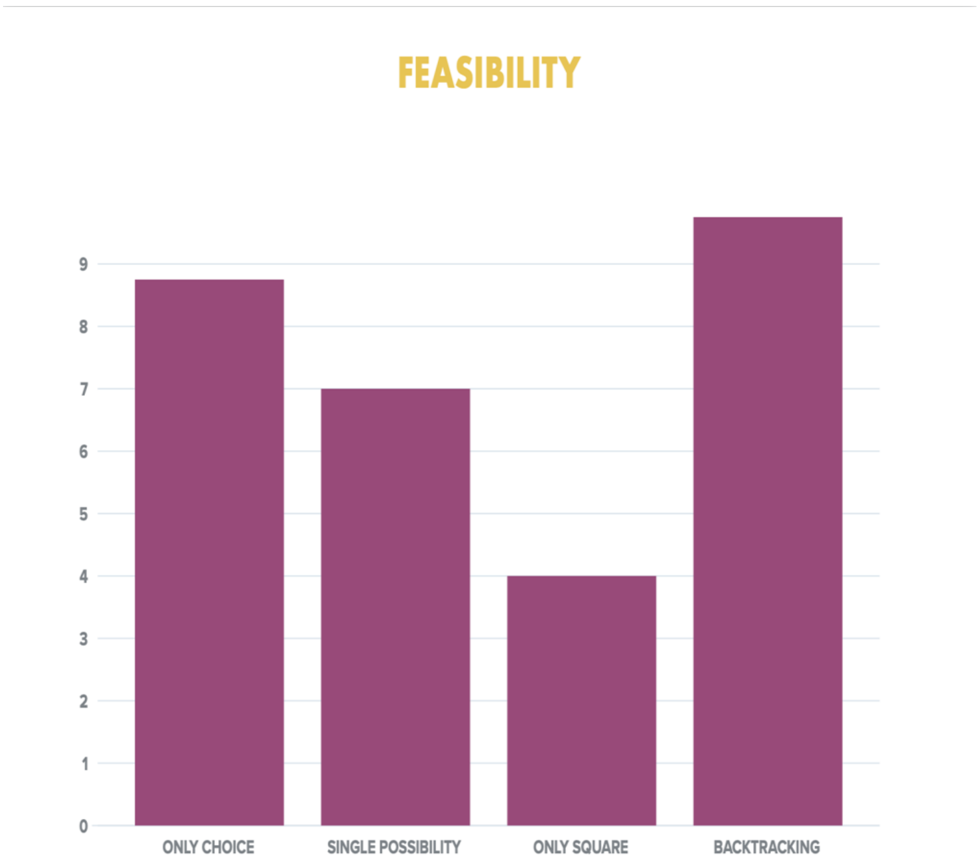
The effect of each method has to be thoroughly checked out so that we can arrive at a suitable conclusion. Effectiveness evaluation uses systematic research from various disciplines to assess how well a program is achieving its objectives. Since every method has a different algorithm we measure their respective effectiveness against each other.

(b) TIME COMPLEXITY



Time complexity of an algorithm signifies the total time required to run a specific process or the entire program until its completion. Here we have considered the time complexity for the various methods that we have taken into account for solving the sudoku puzzle.

(c) FEASIBILITY



Feasibility of a program is the state or degree of being easily or conveniently done. We lookout for the simplest algorithm to find out the quickest solution possible. Since feasibility also shows us which method can be easy to implement practically. Thus an optimal solution can be obtained by inferring the graph.

These measures will help us gain perspective in reaching our goal in the easiest way possible .

3.2 CODE :

```
DATA SEGMENT

BOARD  DB 0,3,2,1,0,0,0,0,0
       DB 4,0,0,0,0,0,0,0,0
       DB 5,0,0,0,0,2,0,0,3
       DB 0,6,7,0,0,3,0,0,9
       DB 0,0,0,8,0,4,0,0,6
       DB 0,0,0,9,0,5,0,0,1
       DB 3,2,1,0,0,6,0,0,5
       DB 0,0,0,0,0,7,0,0,4
       DB 0,0,0,0,0,0,8,3,0,'$'

BOARD2 DB 0,0,0,0,0,0,0,0,0,10
       DB 0,0,0,0,0,0,0,0,0,10
       DB 0,0,0,0,0,0,0,0,0,10
       DB 0,0,0,0,0,0,0,0,0,10
       DB 0,0,0,0,0,0,0,0,0,10
       DB 0,0,0,0,0,0,0,0,0,10
       DB 0,0,0,0,0,0,0,0,0,10
       DB 0,0,0,0,0,0,0,0,0,10
       DB 0,0,0,0,0,0,0,0,0,'$'

DATA ENDS

CODE SEGMENT
ASSUME CS:CODE,DS:DATA
START:
MOV AX,DATA
MOV DS,AX
LEA SI,BOARD
L1:
MOV AL,[SI]
CMP AL,'$'          ;STOP WHEN END OF ARRAY
JZ EN
```

```
CMP AL,0             ;FIND BLANK SPACE
JZ L6
INC SI
JMP L1
L6:
MOV CX,09
L2:
INC AL

CALL CHECK           ;CHECK FOR VALIDITY OF NUMBER
JC L3
JNC L4
L3:
MOV [SI],AL          ;MOVE TO NEXT NUMBER
PUSH AX
DEC CX
PUSH CX
MOV BX,SI
PUSH BX
INC SI
JMP L1
L4:           ;CHECK FOR NEXT NUMBER IF NOT POSSIBLE THEN
             ;BACTRACK
CMP CX,1
JZ L5
LOOP L2
L5:
MOV AH,0
MOV [SI],AH
LH:
POP BX
LEA SI,BOARD
ADD SI,BX
POP CX
POP AX             ;POP OUT OF THE STACK
CMP CX,00
JZ LX
```

```
JNZ L2
LX:
MOV AH,0
MOV [SI],AH
JMP LH          ;JUMP TO LOOP 'H'

EN:             ;PRINTING THE SUDOKU BOARD
MOV CX,81
LEA SI,BOARD
LL:
MOV AL,[SI]
ADD AL,48
MOV [SI],AL
INC SI
LOOP LL
LEA DI,BOARD2
LEA SI,BOARD
L22:
MOV CX,09
OP:
MOV AL,[SI]
MOV [DI],AL
INC SI
INC DI
LOOP OP
MOV AL,[DI]
INC DI
CMP AL,'$'
JNZ L22
LEA SI,BOARD2
LEA DI,BOARD2
MOV AH,09
LEA DX,BOARD2
INT 21H
HLT
```

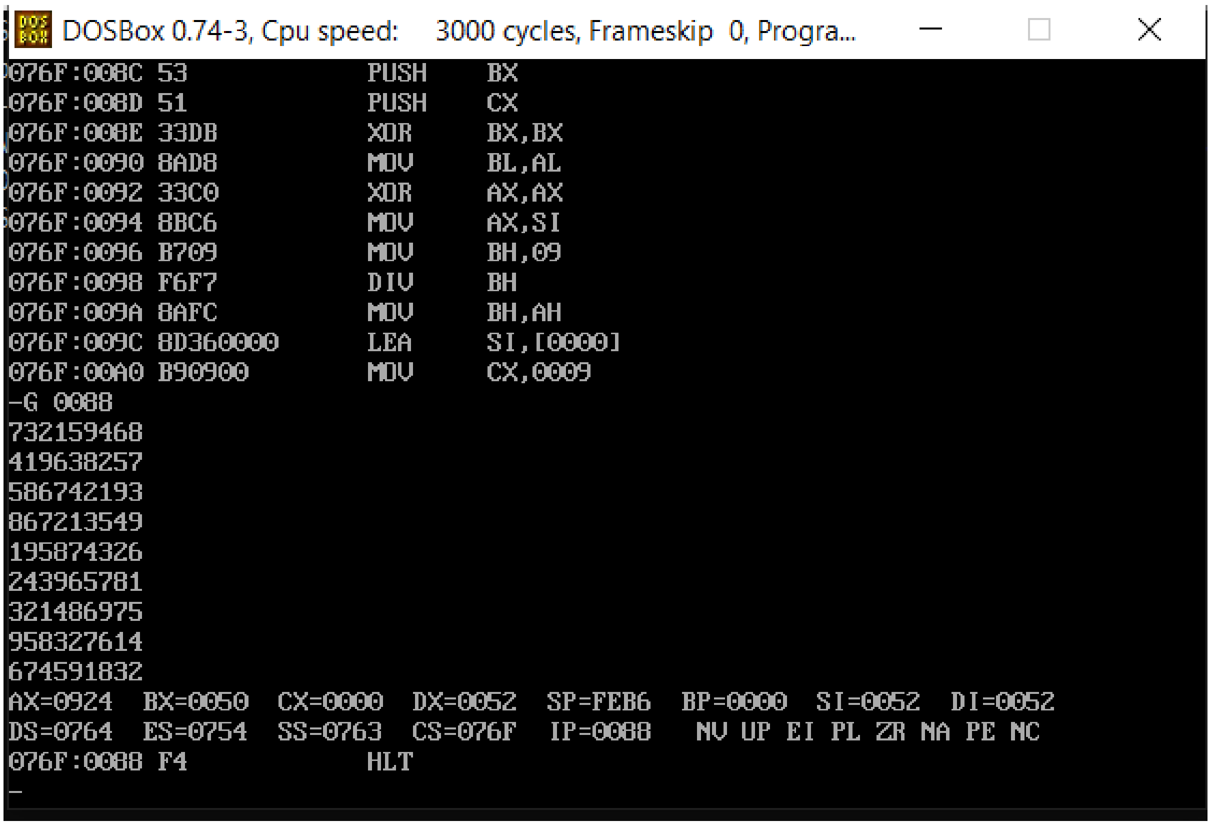
```
CHECK PROC          ;CALL TO CHECK FUNCTION
PUSH AX
MOV BX,SI
PUSH BX
PUSH CX
XOR BX,BX
MOV BL,AL
XOR AX,AX
MOV AX,SI
MOV BH,09
DIV BH
MOV BH,AH
LEA SI,BOARD
MOV CX,09
MOV AH,00
MUL CX
ADD SI,AX
MOV AH,BH
XOR CX,CX
MOV CX,09
L8:
MOV BH,[SI]
CMP BH,BL
JZ L7
INC SI
LOOP L8
MOV AL,AH
MOV AH,00
LEA SI,BOARD
ADD SI,AX
MOV CX,09
L10:
MOV BH,[SI]
CMP BH,BL
JZ L7
ADD SI,09
LOOP L10
```

```
XOR AX,AX
MOV AL,BL
POP CX
POP BX
PUSH BX
PUSH CX
XOR CX,CX
MOV BH,AL
MOV AL,BL
MOV BL,27
PUSH AX
DIV BL
MOV AH,00
MUL BL
POP CX
PUSH AX
MOV AX,CX
MOV BL,09
DIV BL
MOV AL,AH
MOV AH,00
MOV BL,03
DIV BL
MOV AH,00
MUL BL
POP CX
ADD AX,CX
LEA SI,BOARD
ADD SI,AX
MOV BL,[SI]
CMP BH,BL
JZ L7
INC SI
MOV BL,[SI]
CMP BH,BL
JZ L7
INC SI
```

```
MOV BL,[SI]
CMP BH,BL
JZ L7

ADD SI,09
MOV BL,[SI]
CMP BH,BL
JZ L7
DEC SI
MOV BL,[SI]
CMP BH,BL
JZ L7
DEC SI
MOV BL,[SI]
CMP BH,BL
JZ L7

ADD SI,09
MOV BL,[SI]
CMP BH,BL
JZ L7
INC SI
MOV BL,[SI]
CMP BH,BL
JZ L7
INC SI
MOV BL,[SI]
CMP BH,BL
JZ L7


POP CX
POP BX
LEA SI,BOARD
ADD SI,BX
POP AX
STC
```

```
JMP L11
L7:
POP CX
POP BX
LEA SI,BOARD
ADD SI,BX
POP AX
CLC
L11:
RET
CHECK ENDP
HLT
CODE ENDS
END START
```

3.3 SNAPSHOT OF THE OUTPUT



DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...

```
076F:008C 53            PUSH    BX
076F:008D 51            PUSH    CX
076F:008E 33DB          XOR     BX,BX
076F:0090 8AD8          MOV     BL,AL
076F:0092 33C0          XOR     AX,AX
076F:0094 8BC6          MOV     AX,SI
076F:0096 B709          MOV     BH,09
076F:0098 F6F7          DIV     BH
076F:009A 8AFC          MOV     BH,AH
076F:009C 8D360000      LEA     SI,[0000]
076F:00A0 B90900        MOV     CX,0009
-G 0088
732159468
419638257
586742193
867213549
195874326
243965781
321486975
958327614
674591832
AX=0924  BX=0050  CX=0000  DX=0052  SP=FEB6  BP=0000  SI=0052  DI=0052
DS=0764  ES=0754  SS=0763  CS=076F  IP=0088   NV UP EI PL ZR NA PE NC
076F:0088 F4            HLT
-
```

4. CONCLUSION

4.1 CONCLUSION

The objective of the project , i.e., to solve the given SUDOKU PUZZLE , has been fulfilled by comparing the various methods . The solution is found and the various parameters have been optimized . The implementation of all these techniques and strategies has brought the project to its ultimate goal and has reached the final step of its completion .

4.2 INFERENCE

In this project, sudoku solver- it contains several methods and data structures serving as solvers to sudoku games. It can read the board specified by the file at the given path, and importantly, solve all the sudoku puzzles(easy, moderate and hard) in an extremely efficient way.
Any type of sudoku puzzle can be solved provided it is in our standard format of 9x9. The blank spaces are denoted by a 0(null character as said).
The main purpose of the code which is to completely solve sudoku games is fulfilled since-in the game of Sudoku, the player is given a partially-filled 9 × 9 grid, grouped into a 3 × 3 grid of 3 × 3 blocks(sub grids). The objective is to fill each square with a digit from 1 to 9, subject to the requirement that each row, column, and block must contain each digit exactly once. Through generating this Sudoku solver and generator we feel that we have improved our assembly language programming ability. This was perhaps the largest program in terms of time invested and lines of code written that we have created. The problems posed by the project were a good challenge to solve. Writing the solver has demonstrated the advantages of 'smart' algorithms over naïve algorithms evidenced in the measurements above. Finally, we have experienced participating in what could be called a research project.

5. REFERENCES

LIST OF PUBLICATIONS

INTERNATIONAL REFERENCES

- Douglas Samuel Jones "Assembly Programming and the 8086 Microprocessor" , Oxford University Press, 1988
- Thomas P. Skinner, “An Introduction to 8086/8088 Assembly Language Programming**” ,** Wiley publication, 1985
- Arora Barak, “Computational Complexity: A Modern Approach”, Cambridge university Press, 2007
- Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. ,“Introduction to algorithms”, MIT Press, 2009

WEB REFERENCES

- https://en.wikipedia.org/wiki/Sudoku
- https://www.tutorialspoint.com/compile_assembly_online.php
- https://ieeexplore.ieee.org/document/194838
- https://www.researchgate.net/publication/340570563_TEACHING_OF_808886_PROGRAMMING_WITH_8086_ASSEMBLY
- A.K. Ray and K.M.Bhurchandi, “Advanced Microprocessors and peripherals” Tata McGraw Hill publication ,2013

BIODATA

Name	:	Saadhikha Shree S
Mobile Number	:	8870321362
E-mail	:	saadhikhashree.s2019@vitstudent.ac.in
Permanent Address	:	17/22 Dr. Nair Road , Vaishali Apts. , B 10, T-NAGAR , Chennai , Tamil Nadu

Name	:	Rajashri Mahato
Mobile Number	:	7063246463
E-mail	:	rajashri.mahato2019@vitstudent.ac.in
Permanent Address	:	4/9 Sepco Township, B-Zone , Durgapur , West Bengal