

# **NED UNIVERSITY OF ENGINEERING AND TECHNOLOGY**

DEPARTMENT OF COMPUTER SCIENCE & INFORMATION TECHNOLOGY



## **COMPLEX COMPUTING PROBLEM (CCP) PROPOSAL**

*PROJECT TITLE: SUDOKU VALIDATOR*

**COURSE NAME:**

CT-175 (Programming Fundamentals)

**GROUP MEMBERS:**

KHADIJA FASIH (CT-25160)

SYEDA AMNA HASNAIN (CT-25164)

UMME HANI ADEEL (CT-25151)

**GROUP NAME:**

WeedCode

**SUBMITTED TO:**

Mr. MUHAMMAD ABDULLAH

## **1. Project Title:**

Sudoku Validator

## **2. Project Description:**

This project focuses on creating a program that checks whether a completed Sudoku board is valid. Sudoku is a popular  $9 \times 9$  number puzzle where each row, column, and  $3 \times 3$  sub-grid must contain all the digits from 1 to 9 without repeating any number. With this program, users can enter a completed board, and it will automatically verify if all the rules are followed, giving instant feedback on whether the board is valid or not. This removes the potential errors and the difficulty faced while checking manually. This project uses fundamental programming concepts like arrays, loops and conditional statements to automate the validation process of a solved sudoku board.

## **3. Project Methodology:**

- Take a complete sudoku board as input.
- Check each row to make sure any digit from 1-9 is not repeated. If it is repeated, it mark the board as invalid.
- Check each column to make sure any digit from 1-9 is not repeated. If it is repeated, it mark the board as invalid.
- Divide the board into nine  $3 \times 3$  sub-grids. Then check each sub-grid to make sure any digit from 1-9 is not repeated. If it is repeated, it mark the board as invalid.
- If the board follows all the rules, it validates the board.

## **LANGUAGE AND TOOLS:**

Language: C language.

Tool used: Dev C++

## **4. OBJECTIVES:**

The main objective is to design a program which:

- Validate a complete sudoku board.
- Ensures that all the rules are followed and make sure that no numbers are repeated in each row, column and sub-grid.
- Gives clear output and report whether the board is valid or not.

## **5. EXPECTED OUTCOMES:**

- Can accurately determine if the sudoku board is valid or not.
- Simple and user-friendly which gives clear outputs.

## **6. Justification – Why it is a Complex Computing Problem:**

Validating a Sudoku board is considered a complex computing problem because it requires multiple layers of logical verification and efficient data handling. The program must simultaneously ensure three distinct constraints i.e., no repetition in rows, columns, or sub-grids across 81 elements organized in a  $9 \times 9$  grid. This involves the use of nested loops, multi-dimensional arrays, and conditional logic to systematically check every element's relationship with others.

Each value's validity depends on the broader state of the grid, which increases the computational complexity. Moreover, the algorithm demands accuracy, optimized traversal, and immediate error detection to handle invalid cases. The integration of these checks into one cohesive solution highlights the program's ability to manage data validation, iteration, and logical reasoning, making it a complex yet efficient example of problem-solving through programming.