**SUDOKU SOLVER PROJECT**

Name- Mansi Agarwal Semester - IV

Course – B.Tech CSE (DS AND AI)

University Roll no.- 2015176

The project is about taking a partially filled sudoku as an input from the user and giving a completely solved correct sudoku as an output. Sudoku is a 9x9 matrix that has numbers filled from 1-9 such that no row or column contains the same digit twice.

The tools and technology used are as follows:

1. Code Blocks
2. C++ programming language

The project has been made as follows:

1. The first step is creating a new empty file in code blocks and save it with the extension .cpp so as to be able to run the C++ program.
2. Then code is as follows:

#include <iostream>

#define N 9

using namespace std;

int grid[N][N];

bool isPresentInCol(int col, int num)

{

for (int row = 0; row < N; row++)

if (grid[row][col] == num)

return true;

return false;

}

bool isPresentInRow(int row, int num)

{

for (int col = 0; col < N; col++)

if (grid[row][col] == num)

return true;

return false;

}

bool isPresentInBox(int boxStartRow, int boxStartCol, int num)

{

for (int row = 0; row < 3; row++)

for (int col = 0; col < 3; col++)

if (grid[row+boxStartRow][col+boxStartCol] == num)

return true;

return false;

}

void sudokuGrid()

{

for (int row = 0; row < N; row++){

for (int col = 0; col < N; col++){

if(col == 3 || col == 6)

cout << " | ";

cout << grid[row][col] <<" ";

}

if(row == 2 || row == 5){

cout << endl;

for(int i = 0; i<N; i++)

cout << "---";

}

cout << endl;

}

}

bool findEmptyPlace(int &row, int &col)

{

for (row = 0; row < N; row++)

for (col = 0; col < N; col++)

if (grid[row][col] == 0)

return true;

return false;

}

bool isValidPlace(int row, int col, int num)

{

return !isPresentInRow(row, num) && !isPresentInCol(col, num) && !isPresentInBox(row - row%3 ,

col - col%3, num);

}

bool solveSudoku(){

int row, col;

if (!findEmptyPlace(row, col))

return true;

for (int num = 1; num <= 9; num++)

{

if (isValidPlace(row, col, num))

{

grid[row][col] = num;

if (solveSudoku())

return true;

grid[row][col] = 0;

}

}

return false;

}

int main(){

int i,j;

cout<<"\t \*\*\*\* Sudoku Solver \*\*\*\*\t\t\n\n";

cout<<"Enter partially filled sudoku with blank spaces as 0"<<endl;

for(i=0;i<N;i++)

{

for(j=0;j<N;j++)

{

cin>>grid[i][j];

}

}

if (solveSudoku() == true)

{

cout<<"\nThe solved Sudoku is as follows:"<<endl;

sudokuGrid();

}

else

cout << "No solution exists";

}

1. So first of all we include the header files and create a main function where we take the partially filled sudoku as an input from the user wherein the places to be filled are given input as 0.

isPresentInCol( ) is a function which checks whether the number is present int that column or not.

isPresentInRow( ) is a function which checks whether the number is present int that row or not.

isPresentInBox( ) is a function which checks whether the number is present int that 3x3 box or not.

sudokuGrid( ) is a function which prints the output after the sudoku is solved.

findEmptyPlace( ) is a function which checks whether the place is empty or not ( i.e it is filled with 0 or not) and update the rows and columns.

isValidPlace( ) checks if the number if not founf in that row,column and current 3x3 box.

solveSudoku( ) is a function used to check if the Sudoku is completely filled with digits from 1-9 correctly or not and works recursively until the sudoku is completely filled. If the function returns “true” then we get the ouput otherwise a message that “no solution exists” is printed.

1. Now Build the program and Run to get the output which is as follows:

