**Project-1**

…………………………………………………………………………………………………………………………………………………………………..

**Name: Pawan Dhungana**

**Project no: 1**

**Due Date: September 6, 2018**

**Design Document**

**Introduction**

A Sudoku puzzle is a 9x9 grid in which some---usually around 28 or 30---of the cells are filled with integers in the range of 1 to 9; the other cells are empty. A region of the puzzle is one of the nine 3x3 non-overlapping squares of cells within the grid. No integer is duplicated within any row, column, or region, and the puzzle's goal is to fill the remaining cells with integers from 1 to 9 so that this non-duplication property is maintained.

**Data Structures**

The program uses a two-dimensional array to read the 9x9 matrix from a file and reports whether or not the matrix satisfies the requirements of Sudoku; that is, whether or not any value is duplicated in any row, column, or region in the matrix. The program uses ifstream to open and read in the file the user wants to open.

**Functions**

There are 3 functions for reading, opening, testing and printing the result in the screen. They are:  
read\_file(): reads in the file input by the user

Print\_matrix(): prints the matrix and the result whether the given array is a valid sudoku solution or not

isValid(): a Boolean function that checks if the inputted file contains an array with a valid sudoku solution

**Main Program**

In the main program, three different functions are being called. Other than that, the file is being opened in the main program and ifstream is passed through different functions via reference.

**User Document**

A Sudoku puzzle is a 9x9 grid in which some---usually around 28 or 30---of the cells are filled with integers in the range of 1 to 9; the other cells are empty. A region of the puzzle is one of the nine 3x3 non-overlapping squares of cells within the grid. No integer is duplicated within any row, column, or region, and the puzzle's goal is to fill the remaining cells with integers from 1 to 9 so that this non-duplication property is maintained.

.

The program's name is Project1.cpp, to compile and run it, simply enter:

g++ Project1.cpp

a.out

A run of the program might look like this:

Enter the filename you want to open including the extension: m1.dat

7 9 4 6 2 1 3 5 8

5 8 3 4 9 7 2 6 1

2 6 1 3 8 5 4 7 9

4 1 9 7 5 6 8 2 3

6 7 2 8 3 9 1 4 5

3 5 8 2 1 4 7 9 6

8 3 6 5 7 2 9 1 4

9 2 5 1 4 3 6 8 7

1 4 7 9 6 8 5 3 2

Is a Sudoku solution.

**Code Listing:**

#include<iostream>

#include<fstream>

#include<cstdlib>

//function prototypes

using namespace std;

void read\_file(ifstream &in\_f,int matrix[9][9]);

void print\_matrix(int matrix[9] [9], ifstream &in\_f);

bool isValid(int matrix[9][9]);

int main()

{

int matrix[9][9];

cout<<"Enter the filename you want to open including the extension: "<<endl;

ifstream in\_f;

char file[50];//takes input what file user want to open

cin.getline(file,50);

in\_f.open(file);//opens file

read\_file(in\_f,matrix);//calls read\_file function to read the file

print\_matrix(matrix, in\_f);//prints the file

in\_f.close();

if (isValid(matrix)==true)

cout << "Is a sudoku solution. " << endl;

else

cout << "Is not a sudoku solution. " << endl;

in\_f.close();

return 0;

}

void read\_file(ifstream &in\_f,int matrix[9][9])//reads file

{

do{

for (int i = 0; i < 9; i++)

for (int j = 0; j < 9; j++)

in\_f >> matrix[i] [j];

}while(in\_f.good());

}

//Prints out the matrix read from file

void print\_matrix(int matrix[9] [9], ifstream &in\_f)

{

for (int i = 0; i < 9; i++)//for rows

{

for (int j = 0; j < 9; j++)//for columns

cout << matrix[i] [j] << " ";

cout << endl;

}

in\_f.close();

}

bool isValid(int matrix[9][9])//checks if the matrix is valid sudoku solution

{

int i, j;

bool status;

status = true;

for(i=0;i<9;i++){

for (int column = 0; column < 9; column++)//checking if there is a duplication in the column

{

if (column != j && matrix[i] [column] == matrix[i] [j])

status = false;

}

}

for(j=0;j<9;j++)

{

for (int row = 0; row < 9; row++)//checking if there is a duplication in the row

{

if (row != i && matrix[row] [j] == matrix[i] [j])

status = false;

}

}

for (int i = 0; i < 9; i++)

{

for (int j = 0; j < 9; j++)

{

if (matrix[i][j] = 0)

status = false;

}

}

for (int i = 0; i < 9; i++)

{

for (int j = 0; j < 9; j++)

{

if ((matrix[i][j] < 0) || (matrix[i][j] > 9))

status = false;

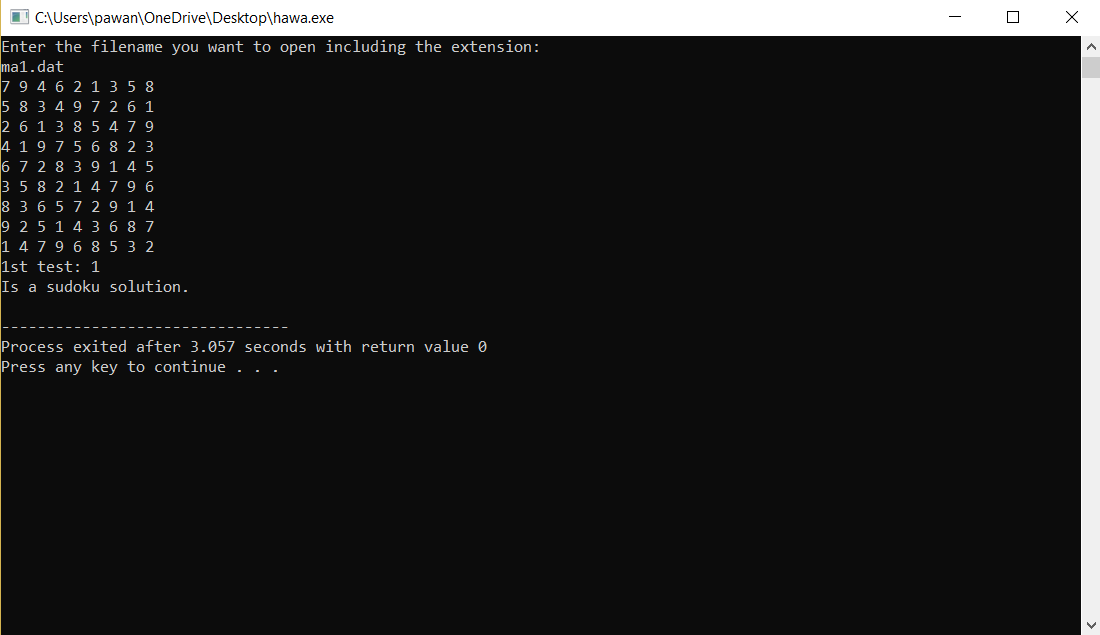
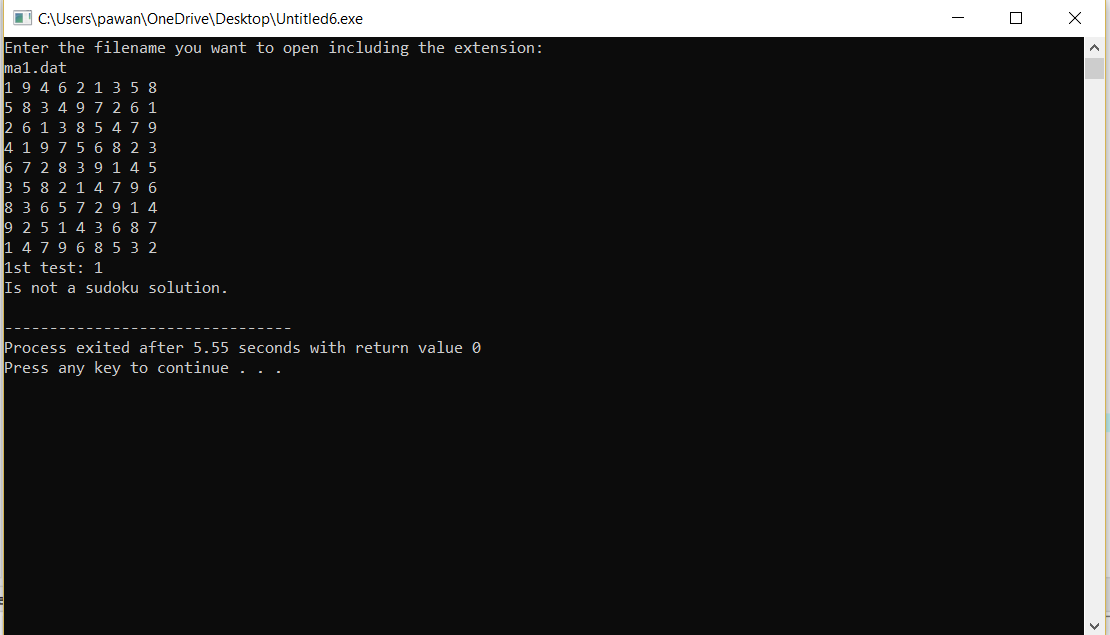
}

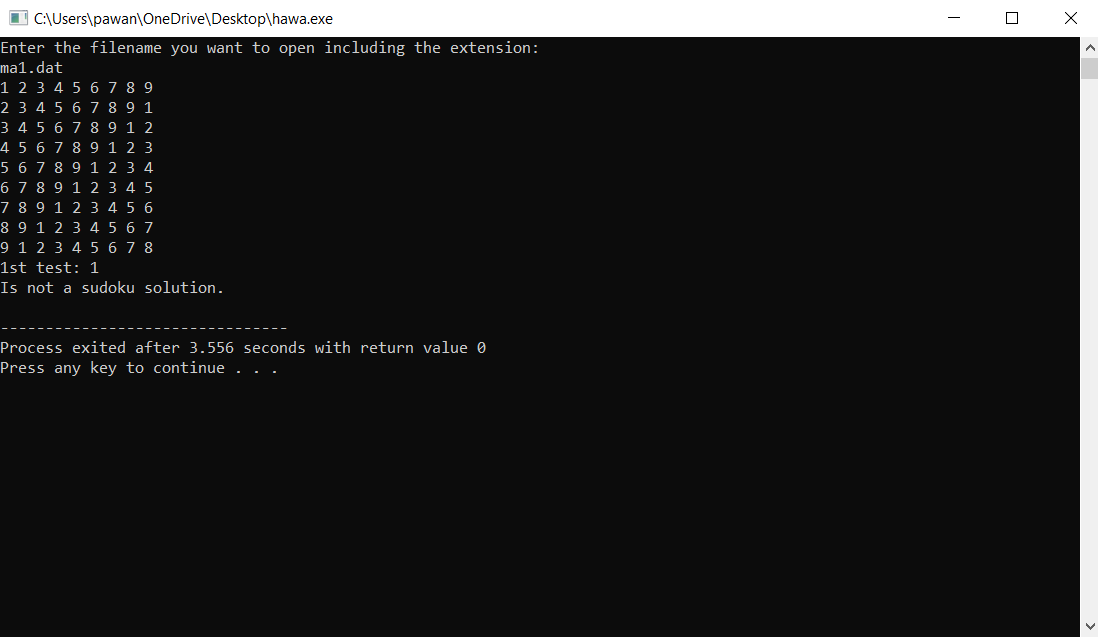
}

return status;

}

**Test Document**

****



**Summary**

In this project, we implemented ifstream to open and read in a 9x9 matrix from a file, read the contents of file, check if the matrix is a valid solution and print out the matrix. We used a two-dimensional array to represent the 9x9 matrix. We checked for different cases like if the elements in the same row/column are different, if a region i.e. nine 3x3 non-overlapping square cells within the grid have different elements without repetition, and if the integers in the files are numbers between 1 and 9.