Sudoku game

Index

Objectives

➢ Situation:

In this module, I am required to write a program which aim at providing an interface for users to play sudoku game. The program should meet the following requirements:

- The program can play by PC users of different ages and different occupations.
- Rules should be defined and stated clearly in the program.
- The program can generate a game board with some of the numbers provided.
- The program can accept inputs from user and modify the content accordingly.
- The program can verify user's solution at the end.

The program should be user-friendly. Users wish to be guided throughout the whole playing process so that they wouldn't meet any insolvable problems. Users may also require input check instantly to avoid mistakes that may lead them to a cul-de-sac. They highly require a smooth procedure when executing the program.

I should provide myself the data source including game board and the related solutions. I Hope my skills in writing program can be enhanced through tackling the task. It is a golden opportunity for me to improve my self-learning skills as well. At the end of this course work, I anticipate a program which can suit the requirements mentioned above can be produced. I also expect to write some extra functions to make the program more perfect. I will put the finished program to web so that Internet user can download it. I hope my program can benefit sudoku players on the Internet.

➢ Sub-problems:

There is no distinct group of users for my program. Every Internet users that is familiar with the rules can use my program already. So there must be users off different abilities. Thus not everyone can solve the same sudoku and choice should be provided. To meet the requirement and solve the problem, I will prepare a number of sudoku games with different difficulties. Every time a game is randomly chosen base on level request by user. If user found that he/she does not want to play that distinct game, he/she can regenerate another one.

Besides, I should consider some common problems usually met by players of sudoku: Sometimes users may get stuck during a game. But it is time-consuming if they really want to go on. Then the smoothness of the program is greatly being affected. To deal with this thorny and unavoidable issue, I will write a function which aims at providing tips to user so that they can keep going. When players have entered wrong numbers into the game board, they may want to delete it. I will write a function which can help users to make this operation become possible.

➢ General requirements:

In order to write the program, I need the following tools:

Tools		Uses
1	A computer	-
2	A program developing software	For writing the program
3	Notepad	For storing contents of suduku game

Analysis

➢ Evaluation:

✧ Solving methods :

There is more than one method to solve this problem. Below are some of the examples with their advantages and drawbacks:

- Write a program using programming languages which run on windows.
  Advantage:
  - ◆ It does not require the connection to the Internet. Thus it can be run on any computer without Internet connection.
  - ◆ It is easy to write and can be run on machine with different platform. Thus it meets demand of efficiency and effectiveness.

  Disadvantage:
  - ◆ The diversification is relatively small compared with other method such as web-based. For instance, the font style, background color or picture. It is more difficult to decorate the interface as lots of studies are required if you want to produce an attractive one.

- Write a web- based program which allow user to play on the Internet.
  Advantage:
  - ◆ User does not need to have the executing program and text files containing sudoku games on the computer. It helps to save space and prevent problems like fail to open those text files.

  Disadvantage:
  - ◆ Web hosting is needed. The one that we can commonly found like Netfirm are not reliable and durable. Your account may be deleted easily. However the reliable one always cost a lot. It is not economical.

- Write the program using Macromedia Flash
  Advantage:
  - ◆ Animations and music can be added to improve the liveliness of the game.

  Disadvantage:
  - ◆ It requires good designing and drawing skills.

I have chosen the first method because:

- This module is about programming but not web authoring.
- I know little about web authoring and hosting and it takes time to acquire knowledge about it. I want to put more time to concentrate on learning one programming language.
- My drawing skill is quite poor. I am not confident in it

✧ Programming language:

After choosing the above method, there is still a great deal of programming languages for me to choose from:

Programming language		Characteristic
1	C/C++	The source code and object code allocate a little space only. It can operate computer at low-level using high level language.
2	Pascal	It is popular in school and designed for teaching structured programming.
3	Microsoft Visual Basic	It can produce a graphical user interface, its function facilitate input and output.
4	Perl	It's execute time is longer as it use interpreter as translator.

I have chosen C to code the program as:

- In my module course, my teacher teaches me to use C. So I am quite familiar with it.
- It is time consuming to learn other type of languages.
- It is portable and can be used in different computers with little modification.

I have also chosen Notepad to produce text files as:

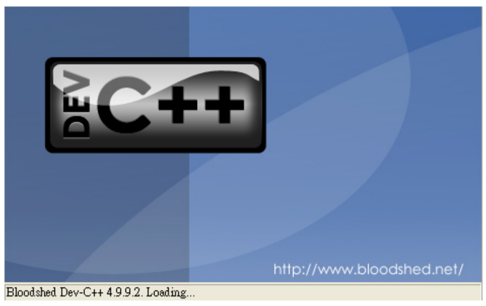
- Notepad is present in every windows, the text file can be modified easily.

✧ Program developing tool:

However, there are also a number of compliers to choose from:

✧ Dev-C++
✧ Borland C++
✧ Visual C++

The first one is my choice because:


DEV C++
http://www.bloodshed.net/
Bloodshed Dev-C++ 4.9.9.2. Loading...

- It is free of charge.
- Its debugging feature helps me a lot. Whenever I come across errors, this software will warn me and indicate where the errors are found. So I can easily carry out debugging work and thus enhance my efficiency.

- All of my classmates use it. We can have discussions and help each other in learning it.

✧ **Choices of interface:**

Type	Characteristics	
Command Line Interface(CLI)	All things are displayed as statements on black windows. Instructions are issued by key in particular commands.	Advantages: It is very efficient foe skilled users.
		Disadvantages: Users need to memorize a set of commands.
Graphical User Interface(GUI)	Information is displayed in multiple ways such as picture and text. Instructions are issued by clicking menu or icons.	Advantages: It is intuitive so it is easy to use.
		Disadvantages: It requires more resources such as memory.

My choice: I have chosen Command Line Interface (CLI) as my program's interface. It is because I think learning is a gradual process, I treat GUI as a higher level skill, if I can not make simple skills through, it will be difficult to have a good performance in adapting higher level interface. So I decide to have a try in using CLI first.

➢ Methods to write out the solution:

With tools and language chosen, I started to design my solution. As my teacher advice us to solve the problem by dividing it into several parts and write functions to implement them, I followed the advice and break the problem into the following 5 parts, the following are them with the possible methods in writing it:

✧ **Generating Sudoku game:**

Possible methods	Advantage & Disadvantage
Store games in text file, open it and read in the contents when running the program.	Advantage: It is relatively easy to write.
	Disadvantage: Game's variation is weak. It takes time to produce text file.
Write codes to generate sudoku game	Advantage: Game's variation is much larger. No text file is needed.
	Disadvantage: It takes more time to construct as the codes involve complex logic.

My choice: I have chosen to read game from a file because I can have better control about the solution. otherwise I may need to write a function to find solution.

✧ **Input method:**

Possible methods	Advantage & Disadvantage
Ask users to key in directly using a keyboard.	Advantage: Uses can own more freedom.
	Disadvantage: Wrong input is usually received.
Make a table with all keys required present and ask users to choose from.	Advantage: Uses can be guided. It is more user- friendly.
	Disadvantage: Displaying of table occupy large space in program.

My choice: I have chosen the former one because need not always refer to the keys in table.

✧ **Instant check for user's input:**

Possible methods	Advantage & Disadvantage
Check against solution I prepared	Advantage: Better control of solution. Easy to write.
	Disadvantage: Alternate solutions may be judged wrongly.
Check against rules	Advantage: It avoid wrong appraisal on alternate solutions.
	Disadvantage: It takes long time to code and code is much longer.

My choice: I have chosen the later one because I don't know whether there is another solution to the game I prepared. It is a failure if I regard alternate solution as wrong one.

✧ **Check whether the game end or not:**

Possible methods	Advantage & Disadvantage
Check if there is blanket present. If 'yes', the game has not end. End for 'no'.	Advantage: Direct and simple to write and code is short.
	Disadvantage: It is clumsy if a two or three dimensional array is used to store the content of game.
First count the number presented in the board, then record how many inputs the user has keyed in. If their sum is 81, end the game, continue for sum less than 81.	Advantage: Need not check game board.
	Disadvantage: Require longer codes.

My choice: I have chosen the first method as I think it is much easier to write.

✧ **Check solution:**

Possible methods	Advantage & Disadvantage
Check against solution I prepared	Advantage: Better control of solution. Easy to write.
	Disadvantage: Alternate solutions may be judged wrongly
Check against rules	Advantage: It avoid wrong appraisal on alternate solutions
	Disadvantage: It takes long time to code and code is much longer.

My choice: I have chosen the later one because it can help to avoid wrong appraise on second solution.

➢ Rules:

Despite I can define the game rules on my own, I decided to adopt the one provided by **www.wikipedia.org** so as to suit different users. It is because the rules are the common one which most player of sudoku familiarize with.

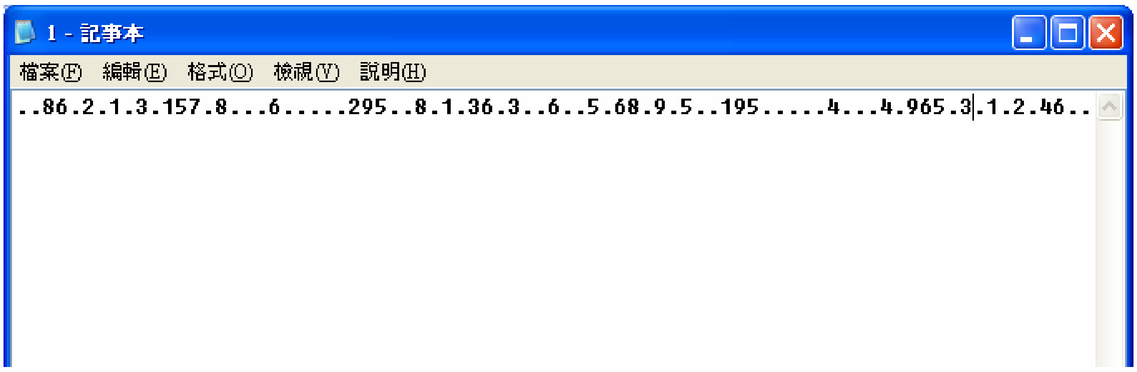
The rule is to fill in all the blankets in the game board according to numbers given so that every column, row and 3 x 3 box contains each of the digits 1 to 9 without repetition.

➢ Input method:

My program mainly accepts two groups of inputs. They are user’s keyboard input and contents of sudoku game read fro text file.

✧ Data storage in text file:

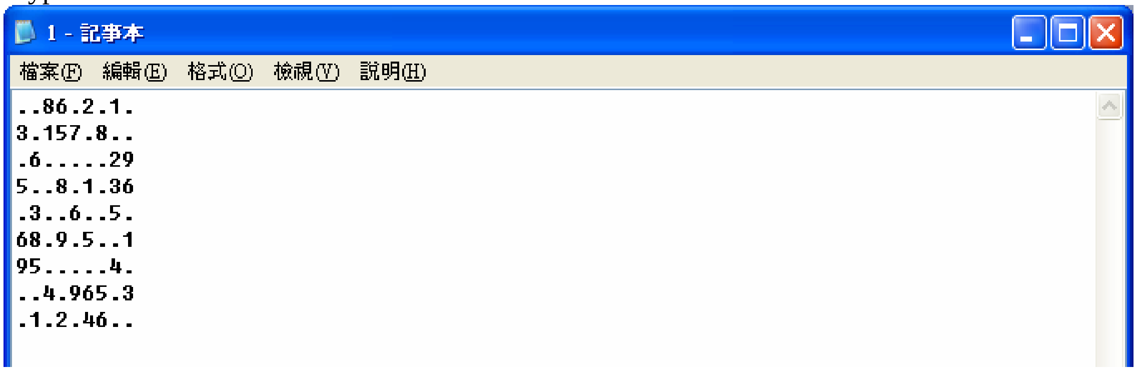
There are a variety of formats used to store game contents in text file:
First type:



```
..86.2.1.3.157.8...6.....295..8.1.36.3..6..5.68.9.5..195.....4...4.965.3.1.2.46..
```

All contents are keyed in as one line without breaking. The non-filled spaces are represented by either dots, 0 or just leave it as space.

Second type:



```
..86.2.1.
3.157.8..
.6.....29
5..8.1.36
.3..6..5.
68.9.5..1
95.....4.
..4.965.3
.1.2.46..
```

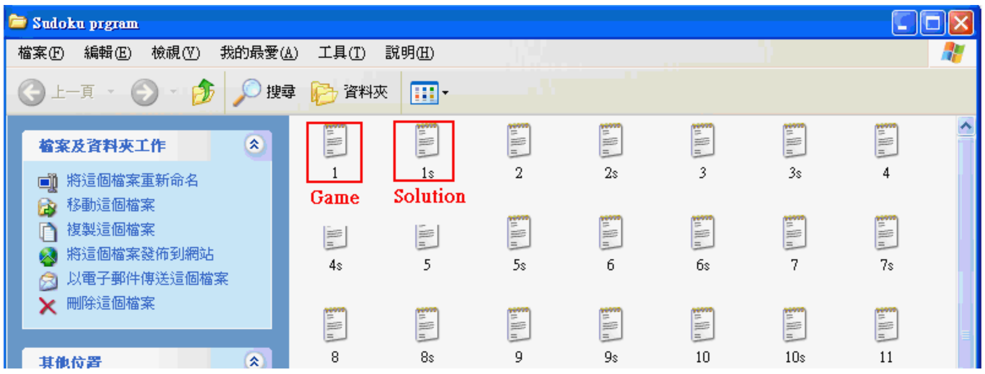
Contents are stored in 9x9 format with breaking. The non-filled spaces are also represented by either dots, 0 or just leave it as space.

I have chosen the second format to gain better management of data. It is easy and clear to read since match with real situation. Store data in such format can help me to eliminate missing of any number because each line is off the same length. If one line is found to be longer or shorter, that line must contain error.

I used dots to represent those non-filled spaces. If I use space, transcription error occurs easily, some place may be missed. Dot is much easier to recognize. If I use 0, 0 will be read in also when running the program. I think it is undesirable.

✧ **Naming of text files:**

I have prepared 30 sudoku games. 1-10 are for level easy, 11-20 are for level medium and the rest is for level hard. Their filename are just 1-30. I have also prepared solutions of them. They are named the same as game except –s is added at the end.



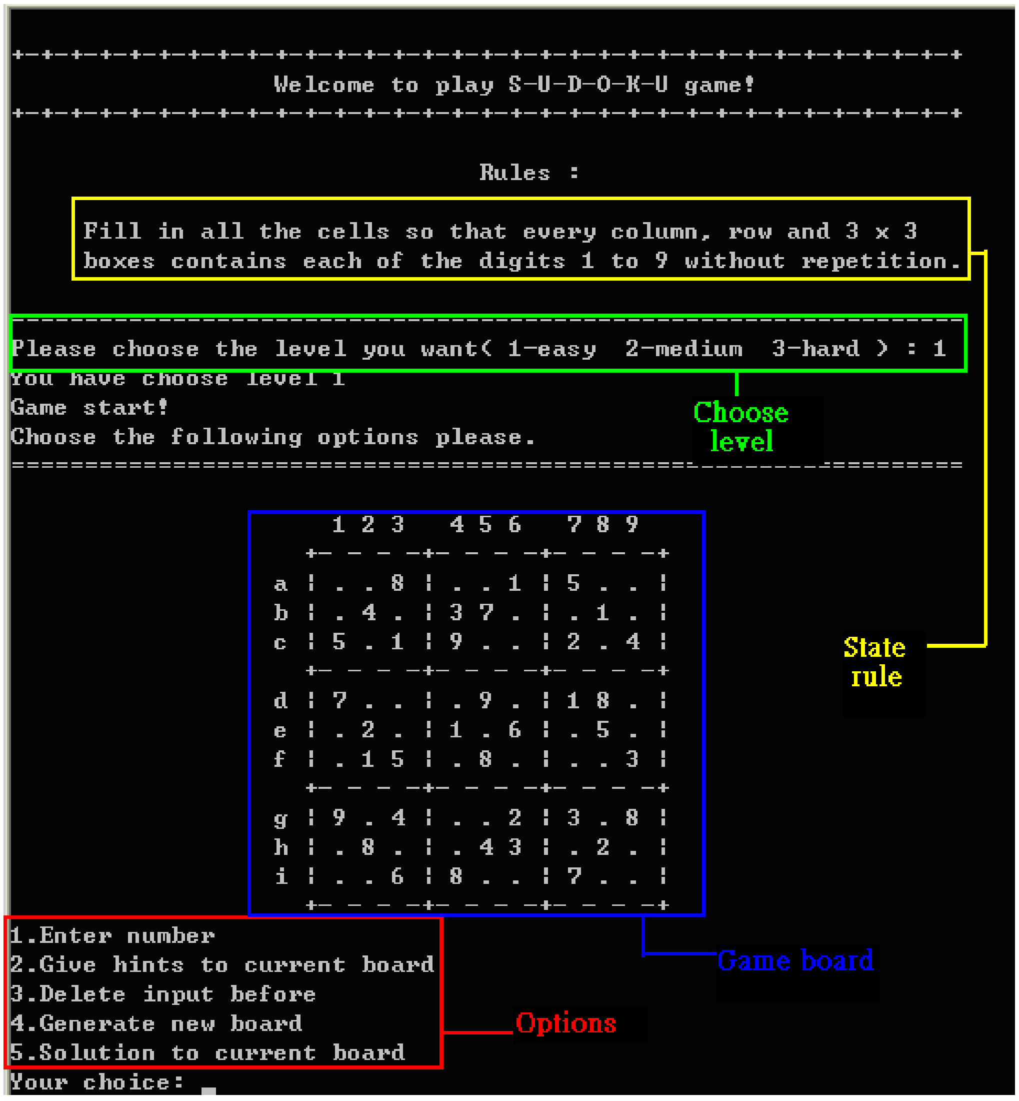
Every time, one file is being opened base on level chosen by user. Users will then input using a keyboard.

➢ Output method:

The major output of my program will all be displayed on the screen instantly when executing the program. No other output like text file will be produced. When solution of player is not correct, the correct answer will be read in and display on the screen.

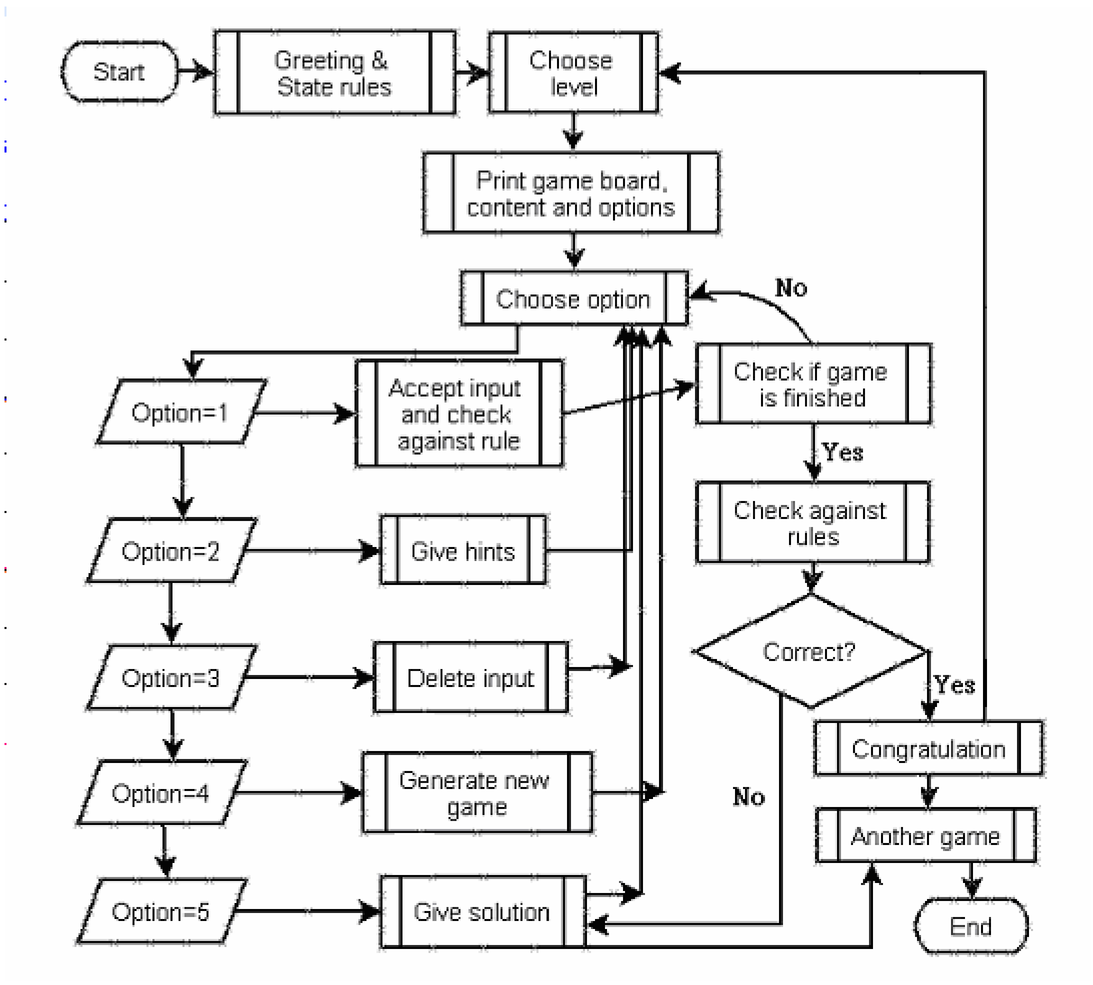
Design & Implementation

This is the overall layout of my program, my program keep going back to the main menu after option 1 to 4 is chosen and completes:.



```
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
                  Welcome to play S-U-D-O-K-U game!
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

                              Rules :

     Fill in all the cells so that every column, row and 3 x 3
     boxes contains each of the digits 1 to 9 without repetition.

----------------------------------------------------------------
Please choose the level you want( 1-easy  2-medium  3-hard ) : 1
You have choose level 1
Game start!
Choose the following options please.
================================================================

                      1 2 3   4 5 6   7 8 9
                    +- - - -+- - - -+- - - -+
                  a ¦ . . 8 ¦ . . 1 ¦ 5 . . ¦
                  b ¦ . 4 . ¦ 3 7 . ¦ . 1 . ¦
                  c ¦ 5 . 1 ¦ 9 . . ¦ 2 . 4 ¦
                    +- - - -+- - - -+- - - -+
                  d ¦ 7 . . ¦ . 9 . ¦ 1 8 . ¦
                  e ¦ . 2 . ¦ 1 . 6 ¦ . 5 . ¦
                  f ¦ . 1 5 ¦ . 8 . ¦ . . 3 ¦
                    +- - - -+- - - -+- - - -+
                  g ¦ 9 . 4 ¦ . . 2 ¦ 3 . 8 ¦
                  h ¦ . 8 . ¦ . 4 3 ¦ . 2 . ¦
                  i ¦ . . 6 ¦ 8 . . ¦ 7 . . ¦
                    +- - - -+- - - -+- - - -+
1.Enter number
2.Give hints to current board
3.Delete input before
4.Generate new board
5.Solution to current board
Your choice: _
```

➢ Flowchart of the program:



From the above flowchart, my program will first greet users [greetintmessage()] and state the rules [rules()] of playing the game. Users then can choose the level [levelchoosing()] of difficulties they want. Afterward, a game board will be printed [printboard()] on the screen with part of the numbers given [generatecontent()].

5 options are then available for chosen. If user's selection is 1, the program will accept input [acceptinput()] from he/she and check both correctness and the validness. Correct one will be filled in and user should choose whether they want to reject invalid one. This process repeat until all grids is being occupied [checkgameend()]. Then my program will check the solution given by user against rules [checkrow()& checkcolumn()]. After checking, he/she will be told where the mistakes are found if solution is not correct. Users can then select to play another game [playothergame()] or just end the program.

Option 2 is written for requesting clues [givehint()] to current board, option 3 can helps to delete input [deletenumber()] by users before. Options 4 and 5 are used to generate another game [changeboard()] off the same level and provide solution [printsolution()] to current board respectively.

✧ Code of main():

```
main()
    {
       int level,option;
       int randnum;
       int crow,ccol;

       greetintmessage();
       rules();
       do{
           level = levelchoosing();
           initboard();
           printf("==============================================================
           ===========\n\n");
           randnum = generatecontent(level);
           do{
               printboard();
               printf("1.Enter number\n2.Give hints to current
               board\n3.Delete input before\n4.Generate new board\n5.Solution
               to current board\n");
               printf("Your choice: ");
               option = verifyoption();
               if(option == '1')
                   {
                       printf("\nPlease choose the grid you want and input
                       number.");
                       acceptinput();
                       system("CLS");
                   }
               else if(option == '2')
                   {
                       givehint(randnum);
                   }
               else if(option == '3')
                   {
                       printf("\nPlease choose the grid you want to delete:");
                       deletenumber(randnum);
                   }
               else if(option == '4')
                   {
                       randnum = generatecontent(level);
                       system("CLS");
                   }
               else
                   {
                       printf("\nCorrect solution is:\n");
                       printsolution(randnum);
                   }
           }while(checkgameend() != 1);

           if(option != '5'){
               printf("Great!You have completed the sudoku!\n");
               printf("Let's check the answers now........\n\n");
               printboard();
               crow = checkrow();
               ccol = checkcolumn();s
               if(crow == 1 && ccol == 1)
                   {
                       congratulation();
```

```
                    }
                else
                    {
                        printf("\nSorry,your solution is not correct.\n");
                        printf("The correct solution should be: \n");
                        printsolution(randnum);
                    }
                }
        }while(playothergame() == 1);
}
```

➢ Data Structure:

There are a number of variables used in my program:

Variable name	Type	Function
int gameboard[9][9]	array	This is the only global variables used in my program. It is used to store the content of the game board throughout the whole process. I choose it instead of gameboard[81] or others because it is more close to the real game board. It facilitate the writing of checking part for determine allowance, especially check 3X3 boxes. I can think of the method easily.
char input[10]	character	Store input by users
int i,j	integer	Used as counter of for-loop
int randnum	integer	Store the name of files randomly generated according to level request from users. Later it is combined with '.txt' to form a filename for opening of file.
char string[10]=".txt"	Character(string)	Used to store '.txt'
char filename[20]	character	Used to store filename of file to be opened
char option[10];	character	Store option entered by user
char readchar	character	Store the character read from a file every time
int startingrow ,startingcol	integer	Store the coordinate of the first grid of a box that the input of users is located in
Int counter,count1, count2	integer	Counters used in functions

➢ Algorithm:

Following are the detail description and implementation of all functions that has existed in my program:

✧ greetintmessage();

Description: This function welcomes the user to use my program.

Input	Nil
Output	Nil

Code:

```
void greetintmessage()
    {
    printf("\n+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+\n");
    printf("%18cWelcome to play S-U-D-O-K-U game!\n",' ');
    printf("+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+\n");
    printf("\n");
}
```

✧ rules();

Description: This function state rules of sudoku game.

Input	Nil
Output	Nil

Code:

```
void rules()
{
   printf("%32cRules :\n",' ');
   printf("\n%5cFill in all the cells so that every column, row and 3 x 3\n",' ');
   printf("%5cboxes contains each of the digits 1 to 9 without repetition.\n",' ');
}
```

✧ levelchoosing();

Description: Ask users to select difficulties by key in 1, 2 or 3 which corresponds to easy, medium or hard respectively. Users are required to input again if it is out of range. Program goes on only when correct input is received. At the end, the request level will be returned.

Input	level
Output	Level[0]

Code:

```
int levelchoosing()
{
   char input[10];
80      printf("\n=============================================================
        ====\n");
81      printf("Please choose the level you want( 1-easy  2-medium  3-hard ) : ");
82      do{
83          scanf("%s",input);
84          if ((strlen(input) != 1) || !( (input[0]<'4') && (input[0]>'0') ))
85              {
86                  printf("Input out of range!Input again please: ");
87              }
88      } while ((strlen(input) != 1) || !( (input[0]<'4') && (input[0]>'0') ));

89      printf("You have choose level %d\n",input[0]-48);
90      printf("Game start!\nChoose the following options please.\n");

91      return input[0];
92  }
```

Line 84 & 88 is a good method to ensure user's input is correct. It checks the length and value at a time.

✧ verifyinput();

Description: This function is used to check validness of user's input in playothergame() and to confirm input in option 1. The user input must be containing one character and ranged from 48 to 50 in terms of ASCLL code value only. Invalid (>2 or <0) one will be rejected and re-input is requested. This verifying method is good that the program won't go panic when sign or English letters are type in. It returns value of input.

Input	Input
Output	Input

Code:

```
91   int verifyinput()
92   {
93       char input[10];
94       do{
95           scanf("%s",input);
96           if ((strlen(input) != 1) || !( (input[0]<'2') && (input[0]>='0') ))
97               {
98                   printf("Input out of range!Input again please: ");
99               }
100      } while ((strlen(input) != 1) || !( (input[0]<'2') && (input[0]>='0') ));

101      return input[0];
102  }
```

✧ initboard();

Description: This function is used to initialize the content of the array gameboard[9][9] to be spaces at the very beginning by means of two for-loops.

Input	Nil
Output	Nil

Code:

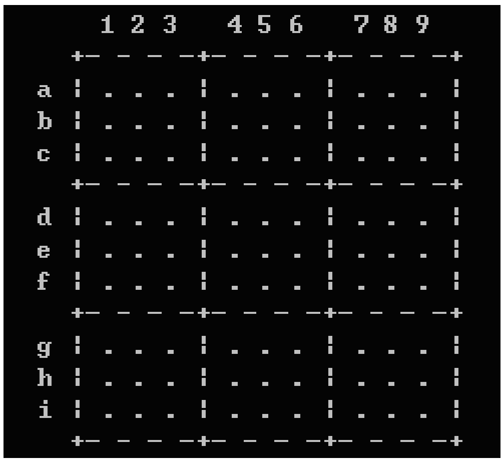
```
103  void initboard()
104  {
105      int i,j;
106      for(i = 0;i < 9;i++){
```

```
107          for(j = 0;j < 9;j++){
108                 gameboard[i][j] = ' ';
109               }
110          }
111 }
```

✧ printboard();

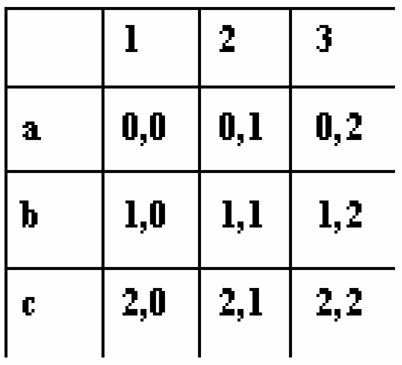
Description: This is a function used to print out the game board on the screen. Columns and rows are called 1 to 9 and a to i respectively. The content is hold by a two-dimensional array.   Each grid means one place in array gameboard[9][9]. E.g. 1a= gameboard[0][0]

Input	Nil
Output	gameboard



```
        1 2 3   4 5 6   7 8 9
      +- - - -+- - - -+- - - -+
    a | . . . | . . . | . . . |
    b | . . . | . . . | . . . |
    c | . . . | . . . | . . . |
      +- - - -+- - - -+- - - -+
    d | . . . | . . . | . . . |
    e | . . . | . . . | . . . |
    f | . . . | . . . | . . . |
      +- - - -+- - - -+- - - -+
    g | . . . | . . . | . . . |
    h | . . . | . . . | . . . |
    i | . . . | . . . | . . . |
      +- - - -+- - - -+- - - -+
```

The layout of my game board



	1	2	3
a	0,0	0,1	0,2
b	1,0	1,1	1,2
c	2,0	2,1	2,2

The relative position of each grid

Code:

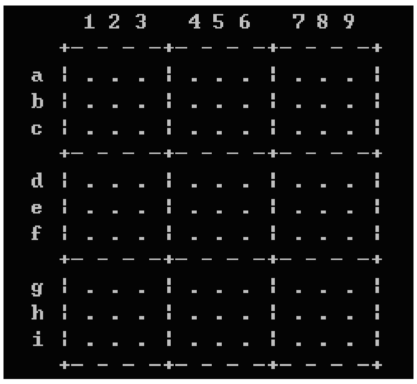
```
112 void printboard()
113 {
114    char space = ' ';
115    char separator[] = "+- - - -+- - - -+- - - -+";
116    char row = 'a';
117    int i, j;

118    printf("%20c  1 2 3   4 5 6   7 8 9\n",space);
119    for (i=0;i<9;i++)
120        {
121             if(i%3 ==0)
122              {
123                  printf("%20c%s\n",space,separator);
124              }
125              printf("%18c%c | %c %c %c | %c %c %c | %c %c %c
             |\n",space,row+i,gameboard[i][0],gameboard[i]
             [1],gameboard[i][2],gameboard[i][3],gameboard[i]
             [4],gameboard[i][5],gameboard[i][6],gameboard[i]
             [7],gameboard[i][8]);
126        }
127    printf("%20c%s\n",space,separator);
128 }
```

I define a variable for separator and space in line 114 & 115 to prevent writing it for many times in the code. It really helps to save lines.

✧ generatecontent();

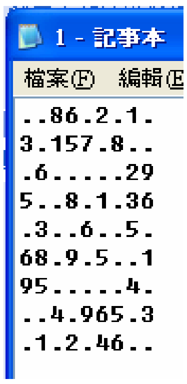
Description: It receives the value of level from level choosing. The function then randomly generates a number called randnum. I use number to be the files name. The randnum then combine with a string containing .txt. The outcome is then used to open the distinct file. Contents inside will be read one by one into the array gameboard[9][9]. It returns value of randnum.

Input	level
Output	Content of a certain file will be displayed on game board



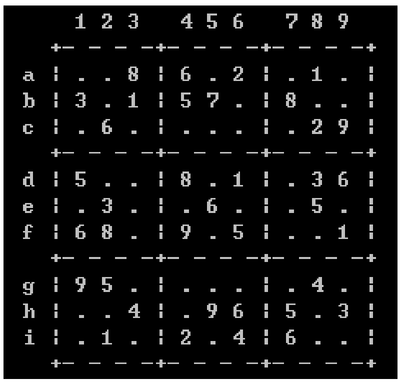
```
    1 2 3   4 5 6   7 8 9
  +- - - -+- - - -+- - - -+
a ¦ . . . ¦ . . . ¦ . . . ¦
b ¦ . . . ¦ . . . ¦ . . . ¦
c ¦ . . . ¦ . . . ¦ . . . ¦
  +- - - -+- - - -+- - - -+
d ¦ . . . ¦ . . . ¦ . . . ¦
e ¦ . . . ¦ . . . ¦ . . . ¦
f ¦ . . . ¦ . . . ¦ . . . ¦
  +- - - -+- - - -+- - - -+
g ¦ . . . ¦ . . . ¦ . . . ¦
h ¦ . . . ¦ . . . ¦ . . . ¦
i ¦ . . . ¦ . . . ¦ . . . ¦
  +- - - -+- - - -+- - - -+
```

+



```
..86.2.1.
3.157.8..
.6.....29
5..8.1.36
.3..6..5.
68.9.5..1
95.....4.
..4.965.3
.1.2.46..
```

=



```
    1 2 3   4 5 6   7 8 9
  +- - - -+- - - -+- - - -+
a ¦ . . 8 ¦ 6 . 2 ¦ . 1 . ¦
b ¦ 3 . 1 ¦ 5 7 . ¦ 8 . . ¦
c ¦ . 6 . ¦ . . . ¦ . 2 9 ¦
  +- - - -+- - - -+- - - -+
d ¦ 5 . . ¦ 8 . 1 ¦ . 3 6 ¦
e ¦ . 3 . ¦ . 6 . ¦ . 5 . ¦
f ¦ 6 8 . ¦ 9 . 5 ¦ . . 1 ¦
  +- - - -+- - - -+- - - -+
g ¦ 9 5 . ¦ . . . ¦ . 4 . ¦
h ¦ . . 4 ¦ . 9 6 ¦ 5 . 3 ¦
i ¦ . 1 . ¦ 2 . 4 ¦ 6 . . ¦
  +- - - -+- - - -+- - - -+
```

Concepts of how content is generated

Code:

```
int generatecontent(level)
{
    int i,j,randnum;
    char readchar,filename[20],string[10] = ".txt";
    FILE *fp;
    srand(time(NULL));

    if(level == '1')
         {
            randnum = (rand() %10) + 1;
            sprintf(filename,"%d%s",randnum,string);
         }
    else if(level == '2')
         {
            randnum = (rand() %10) + 11;
            sprintf(filename,"%d%s",randnum,string);
         }
    else if(level == '3')
         {
            randnum = (rand() %10) + 21;
            sprintf(filename,"%d%s",randnum,string);
149          }

150    fp=fopen(filename,"r");
151    if(fp==NULL)
152        {
153            printf("cannot open the file!\n");
154            exit(1);
155        }

156    readchar = fgetc(fp);
157    while(readchar != EOF) {
158        for(i=0;i<9;i++)
159            {
160               for(j=0;j<=9;j++)
161                    {
162                        gameboard[i][j] = readchar;
163                        readchar = fgetc(fp);
164                    }
165            }
166    }
167    fclose(fp);

168    return randnum;
169 }
```

I use sprintf() to combine the randnum(filename) with .txt for opening in line 138, 143 & 148 as there are three levels.

✧ verifyoption();

Description: Verify the correctness of option inputted. It works the same as **verifyinput()** **except** the range is 1 to 5.

Input	Option
Output	Option

Code:

```
170 int verifyoption(void)
171
172    char option[10];
173    do{
174        scanf("%s",option);
175        if ((strlen(option) != 1) || !( (option[0]<'6') && (option[0]>'0') ))
176            {
177                printf("Input out of range!Input again please: ");
178            }
179    }while((strlen(option) != 1) || !( (option[0]<'6') && (option[0]>'0') ));
180    return option[0];
181
```

✧ acceptinput();

Description: It accepts input from user. User enter column first followed by row and last is number to be filled in. The function first check correctness of input and number. Then if the input is valid and grid being selected is not yet filled. The array will be modified and new number is added. Otherwise it will be rejected.

The most important part of this function is it helps to check if the number is allowed to be put in instantly. It compares the number with others in the same row, column and 3X3 box. Warning message will be stated if repetitions are found. User can choose not to enter.

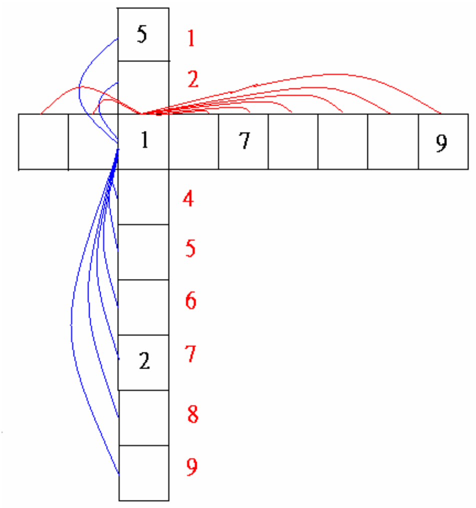
Input	Row, column, number
Output	Valid number is stored in array gameboard[9][9]

Code:

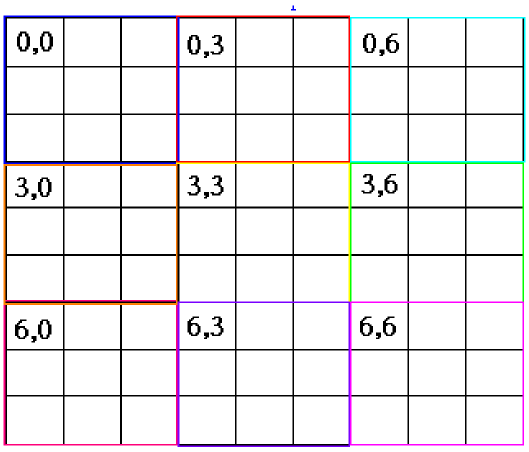
```
void acceptinput()
{
   int i , j,row, col, number;
   char input[20];
   int count1 = 0,count2 = 0;
   int startingrow,startingcol;

   do{
      col = entercolumn() - 49;
      row = enterrow() - 97;

      if (gameboard[row][col] !='.')
         {
            printf("The cell is already occupied!\n");
            printf("Input again please.");
         }
   }while(gameboard[row][col] !='.');

   printf("Which number(1-9)do you want to type in? ");
   do {
      scanf("%s",input);
      if ((strlen(input) != 1) || !( (input[0]<='9') && (input[0]>'0') ))
         {
            printf("Input out of range! Input again please: ");
         }
   } while ((strlen(input) != 1) || !( (input[0]<='9') && (input[0]>'0') ));
   number = input[0];

   for(i = 0;i < 9;i++)
      {
         if(gameboard[row][i] == number)
            {
               count1++;
            }
         if(gameboard[i][col] == number)
            {
               count1++;
            }
      }

   if(row < 3)
      {
         startingrow = 0;
      }
   else if((row > 2) && (row < 6))
      {
         startingrow = 3;
      }
   else if((row > 5) && (row < 9))
      {
         startingrow = 6;
      }
   if(col < 3)
      {
         startingcol = 0;
      }
   else if((col > 2) && (col < 6))
      {
         startingcol = 3;
      }
   else if((col > 5) && (col < 9))
      {
```

```
240             startingcol = 6;
241         }
242     for(i = 0;i < 3;i++)
243         {
244             for(j = 0;j < 3;j++)
245                 {
246                     if(gameboard[startingrow+i][startingcol+j] == number)
247                         {
248                             count2++;
249                         }
250                 }
251         }

252     if((count1 == 0) && (count2 == 0))
253         {
254             gameboard[row][col] = number;
255         }
256     else
257         {
258             printf("==========================================================
                ========\n");
259             printf("Sorry,input number is repeated either in row,column or 3x3
                box\n");
260             printf("Do you really want to input(1-yes 0-no)? ");
261             input[0] = verifyinput();
262             if(input[0] == '1' )
263                 {
264                     gameboard[row][col] = number;
265                 }
266         }

267     count1 = 0;
268     count2 = 0;

269 }
```

Line 207-217 are used to check if same number as user input is appeared on either same row, column. It works in this way:



For example, 1 is user's input. My program compare element in grid near to 1 with user input followed by the sequence. Counter increase by 1 if they are the same. Row works the same. A valid number should result in counter equal to 0.
Line 217-251 are used to check if same number as user input is appeared on same 3x3 boxes. My program first locates which 3x3 box the grid chosen by user is lying on.



The respective 9 3x3 boxes and position of their first place

Using concept of range and data received from user, it can be easily located. Afterward, my program compares the rest elements on the same box with user input. The same, a valid number should result in counter equal to 0. If both counters equal to 0, I can conclude that it is a possible solution.

✧ enterrow();

Description: This function ask user's input of row and helps to check correct length and correct range (a to i) of it.

Input	Row
Output	Nil

Code:

```
int enterrow(void)
{
   char input[10];
   printf("Which row do you want to choose(e.g.a)? ");
            do {
                scanf("%s",input);
                if  ((strlen(input)  !=  1)  ||  !(  (input[0]<'j')  &&
                (input[0]>='a') ))
                   {
                      printf("Input out of range! Input again please: ");
                   }
            }  while  ((strlen(input)  !=  1)  ||  !(  (input[0]<'j')  &&
            (input[0]>='a') ));
   return input[0];
}
```

✧ entercolumn();

Description: This function ask user's input of column and helps to check correct length and correct range (1 to 9) of it

Input	column
Output	Nil

Code:

```
int entercolumn(void)
{
   char input[10];
   printf("\nWhich column do you want to choose(e.g.1)? ");
             do {
                scanf("%s",input);
                if ((strlen(input) != 1) || !( (input[0]<='9') && (input[0]>'0') ))
                    {
                       printf("Input out of range! Input again please: ");
                    }
             } while ((strlen(input) != 1) || !( (input[0]<='9') && (input[0]>'0') ));
   return input[0];
}
```

✧ givehint();

Description: This function reveals the answer of a particular grid according to user’s demand. It works by open the solution file of to the current game. Characters are read in but only the asked grid will be given to player. It has to receive value of randnum. Tips wouldn’t be given if number is presented already.

Input	Row, column
Output	Requested number given

Code:

```
void givehint(randnum)

   int i,j,row,col;
   char space = ' ';
   char readchar,filename[20],string[10] = "s.txt";
   FILE *fp;

   col = entercolumn() - 49;
   row = enterrow() - 97;
   sprintf(filename,"%d%s",randnum,string);
   fp=fopen(filename,"r");
   if(fp==NULL)
      {
          printf("cannot open the file!\n");
          exit(1);
      }
   readchar = fgetc(fp);
   while(readchar != EOF) {
      for(i=0;i<9;i++)
          {
            for(j=0;j<=9;j++)
                {
                    if(i ==row && j == col)
                        {
                             if(gameboard[row][col] == '.')
                                 {
                                     gameboard[i][j] = readchar;
                                 }
                             else
                                 {
                                     printf("==============================
                                             ===================================\n"
                                             );
                                             printf("%23c###################\n",sp
                                             ace);
                                             printf("%23cNumber                is
                                             presented!\n",space);
                                             printf("%23c###################\n",sp
                                             ace);
                                        }
                              }
                    readchar = fgetc(fp);
                }
        }
    }
    fclose(fp);
}
```

✧ deletenumber();

Description: This function helps to delete former input in the game board. If the number is provided to user. It can not be removed. It works in the following way:
1: Open the text file containing the current game.
2: Check whether the grid entered by user is embedded with number or not.
3: If number is presented, the number will not be deleted and vice versa

Input	Row, column
Output	Request number deleted

Code:

```
void deletenumber(randnum)
{
    int i,j,row,col;
    char space = ' ';
    char readchar,filename[20],string[10] = ".txt";
    FILE *fp;

    col = entercolumn() - 49;
    row = enterrow() - 97;
    sprintf(filename,"%d%s",randnum,string);
    fp=fopen(filename,"r");
    if(fp==NULL)
        {
            printf("cannot open the file!\n");
            exit(1);
        }
    readchar = fgetc(fp);
    while(readchar != EOF) {
        for(i=0;i<9;i++)
            {
               for(j=0;j<=9;j++)
                    {
                        if(i ==row && j == col)
                            {
                                if(readchar == '.' && gameboard[row][col] != '.')
                                    {
```

```
                        gameboard[row][col] = '.';
                        printf("=================================================================\n");
                    }
                    else if(readchar == '.' && gameboard[row][col] == '.')
                    {
                        printf("=================================================================\n");
                        printf("%21c######################\n",space);
                        printf("%21cNothing can be deleted!\n",space);
                        printf("%21c######################\n",space);
                    }
                    else
                    {
                        printf("=================================================================\n");
                        printf("%8c#########################################################\n",space);
                        printf("%8cThe number can't be cancelled.It is provided originally!\n");
                        printf("%8c#########################################################\n",space);
                    }
                }
                readchar = fgetc(fp);
            }
        }
    }
    fclose(fp);
}
```

✧ checkgameend();

Description: It checks if the user has finished play the current game or not by counting grid filled with number. The game end if the array is fully filled and 1 will be returned. Otherwise return 0 and game continue.

Input	Nil
Output	Nil

Code:

```
int checkgameend()

    int i,j;
    int counter = 0;
    for(i = 0;i < 9;i++){
        for(j = 0;j < 9;j++){
            if(gameboard[i][j] != '.'){
                counter ++;
```

```
                        }
                    }
             }

  if(counter == 81)
       {
           return 1;
       }
  else
       {
           return 0;
       }
}
```

✧ **checkrow();**

Description: It compares the value of each number in each row with the rest elements when game ended. Element in the first place needed to be compared with rest 8 elements. While that of in second place need to be compared with rest 7 elements only and so on and so forth. Totally 324 comparison will be done. Comparisons are done by function comparerow(). Rows with repetitions will be indicated wordily. Return 1 when no repetition is appeared. Otherwise 0 is returned.

Input	Nil
Output	Nil

Code:

```
int checkrow(void)
{
   int i,j;
   int sum,counter1 = 0,counter2 = 0,counter3 = 0;
   for(i = 0;i < 9;i++)
       {
                      Sum =(gameboard[i][0]+gameboard[i][1]+gameboard[i][2]+
                             gameboard[i][3]+gameboard[i][4]+gameboard[i][5]+gam
                             eboard[i][6]+gameboard[i][7]+gameboard[i][8])-48*9;
           if(sum == 45)
               {
                   counter1 ++;
               }
       }

   for(i = 0;i < 9;i++){
           for(j = 1;j < 9;j++){
           counter3 = comparerow(i,j);
           counter2 += counter3;
           counter3 = 0;
           }
   }
   if((counter1 == 9) && (counter2 == 324))
   {
           return 1;
       }
   else
       {
           return 0;
       }
}
```

✧ comparerow();

Description: This function is being called in checkrow(). It compares the value of each number in each row with the rest elements. Its return value will be less than 8 when there is repetition found in same row. Besides, it also informs user where repetitions are found.

Input	Nil
Output	Nil

Code:

```
int comparerow(i,a)
{
   int j,counter = 0;
   for(j = a;j < 9;j++)
      {
         if(gameboard[i][a-1] != gameboard[i][j])
               {
                  counter ++;
               }
         else
               {
                  printf("More than one %c are found in row
                  %d!\n",gameboard[i][a-1],i+1);
               }
      }
   return counter;
}
```

✧ checkcolumn();

Description: It works the same as checkrow() but it is used to compare numbers in each column instead. Comparison are done by function comparecolumn().

Input	Nil
Output	Nil

Code:

```
int checkcolumn(void)
{
   int i,j;
   int sum,counter1 = 0,counter2 = 0,counter3 = 0;
   for(i = 0;i < 9;i++)
      {
         sum =(gameboard[0][i]+gameboard[1][i]+gameboard[2][i]+
         gameboard[3][i]+gameboard[4][i]+gameboard[5][i]+ga
         meboard[6][i]+gameboard[7][i]+gameboard[8][i])-48*9;
         if(sum == 45)
            {
               counter1 ++;
            }
      }

   for(i = 0;i < 9;i++){
         for(j = 1;j < 9;j++){
         counter3 = comparecolumn(i,j);
         counter2 += counter3;
```

```
            counter3 = 0;
            }
    }
                if((counter1 == 9) && (counter2 == 324))
        {
            return 1;
        }
    else
        {
            return 0;
        }
}
```

➢ comparecolumn();

Description: It works the same as comparerow(), it compares the value of each number in each column with the rest elements. Its return value will be less than 8 when there is repetition found in same row. Besides, it also informs user where repetitions are found.

Input	Nil
Output	Nil

Code:

```
int comparecolumn(i,a)

    int j,counter = 0;
    for(j = a;j < 9;j++)
                {
                    if(gameboard[a-1][i] != gameboard[j][i])
                            {
                            counter ++;
                            }
                    else
                            {
                            printf("More  than  one  %c  are  found  in  column
                            %d!\n",gameboard[a-1][i],i+1);
                            }
                }
    return counter;
}
```

✧ printsolution();

Description: It prints out solution of current game if answer given by user is not correct. It will also be called is user choose option 5.
Principal: Open the solution and read in. Store it in array gameboard[9][9] and print out on screen.

Input	randnum
Output	Solution on board

Code:

```
void printsolution(randnum)
{

    int i,j;
    char readchar,filename[20],string[10] = "s.txt";
    FILE *fp;
```

```

   sprintf(filename,"%d%s",randnum,string);
   fp=fopen("tests.txt","r");
   if(fp==NULL)
       {
           printf("cannot open the file!\n");
           exit(1);
       }
   readchar = fgetc(fp);
   while(readchar != EOF) {
       for(i=0;i<9;i++)
           {
              for(j=0;j<=9;j++)
                   {
                       gameboard[i][j] = readchar;
                       readchar = fgetc(fp);
                   }
           }
   fclose(fp);
   printboard();
}
```

✧ congratulation();

Description: It felicitate user when his/her answer is correct.

Input	Nil
Output	Nil

Code:

```
void congratulation()

   char space = ' ';
   printf("%8c++++++++++++++++++++++++++++++++++++++++++++++++++++\n",space)
   ;
   printf("%8cCONGRATULATION!!You have done the sudoku correctly!\n",space);
   printf("%8c++++++++++++++++++++++++++++++++++++++++++++++++++++\n",space)
   ;

```

➢ playothergame();

Description: It ask user if he/she want another game or not. 1 for continue, 0 for quit.

Input	Nil
Output	Nil

Code:

```
int playothergame()
{
   char input[10];
   printf("==============================================================
   =\n");
    printf("Do you want to play another game(1-yes 0-no)? ");
   input[0] = verifyinput();
   if(input[0] == '1')
      {
         return 1;
      }
   else if(input[0] == '0')
      {
         return 0;
      }
}
```

Testing & Evaluation

➢ Testing plan:

Having the program written, to ensure the program can run correctly is much more important. I tested the correctness of my program with the following procedure:

STEP 1	Test if the program response in a correct manner when receive input from user		
	STEP 1.1	Try to type in different type of wrong inputs	
		STEP 1.1.1	Try inputs that are out of range
		STEP 1.1.2	Try inputs that are not in the required type (e.g. ask for number but enter sign)
	STEP 1.2	Key in correct type of input	
Expectation: When incorrect input is keyed in, the program should ask user to input again until right on is received. Wrong inputs (e.g. signs) wouldn't make the program go panic,			
STEP 2	Test response of program when user wants to fill in a number to a particular grid		
	STEP 2.1	Enter invalid number	
		STEP 2.1.1	The number is presented in the same row
		STEP 2.1.2	The number is presented in the same column
		STEP 2.1.3	The number is presented in the same 3x3 box.
	STEP2.2	Enter valid number	
	STEP2.3	Enter number to grids occupied already	
Expectation: When invalid number is inputted, the program should not accept it immediately. Instead it warns users. The one that do not violate the rules is accepted. Grids that contain number already do not accept input any more.			
STEP 3	Test correctness of function give hints		
	STEP 3.1	Choose grid with number presented already	
	STEP 3.2	Choose grid without number	
Expectation: No clue is give to grid with number.			
STEP 4	Test correctness of function delete number		
	STEP 4.1	Choose grid with number given at the beginning	
	STEP 4.2	Choose grid without any numbers	

	STEP 4.3	Choose grid with number inputted by users	
Expectation: Numbers provided by program cannot be cancelled. Warning message is shown when blanket is being chosen.			
STEP 5	Test whether judgment on user’s solution at the end of game is correct		
	STEP 5.1	Give the program a set of wrong answer	
		STEP 5.1.1	The solution got some mistakes on some rows
		STEP 5.1.2	The solution got some mistakes on some columns
		STEP 5.1.3	The solution got some mistakes on some 3x3 boxes
	STEP 5.2	Give the program a set of correct answer	
Expectation: When wrong set of solution is provided by user. The program shows where the mistakes are found and tell users. Right set receives congratulation message.			

➢ Testing procedure(by myself):

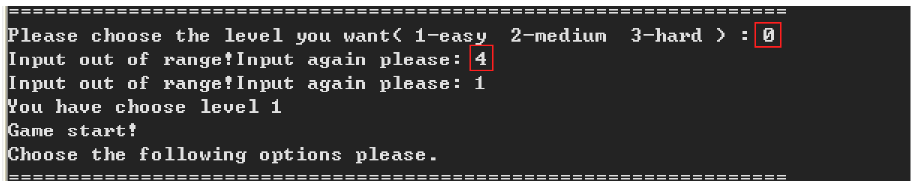
✧ STEP 1:

I must identify sites that require user’s input first:
1: Choose level
2: choose option.
3: Fill in number to the game board (row, column, number)
4: Withdraw wrong input.
5: User wants to have hints from program (row, column, number)
6: User wants to have delete number entered before (row, column, number)
7: Choose whether he/she is going to play another game.

Choose level

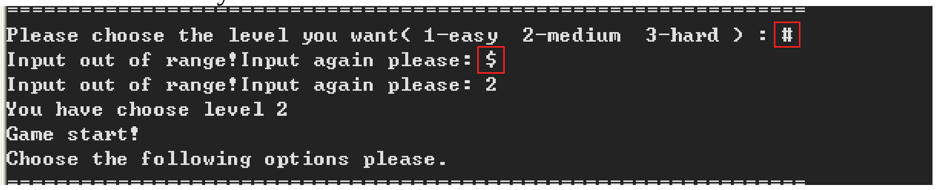
Required input	Required range
Level	1 - 3

Result:



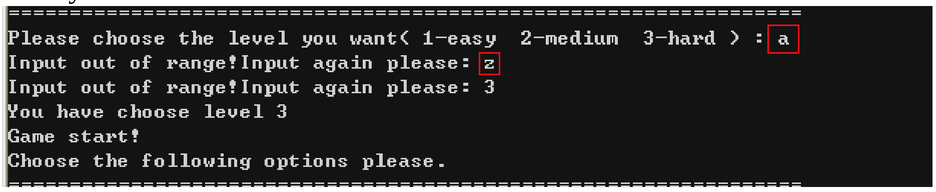
```
=================================================================
Please choose the level you want( 1-easy  2-medium  3-hard ) : 0
Input out of range!Input again please: 4
Input out of range!Input again please: 1
You have choose level 1
Game start!
Choose the following options please.
=================================================================
```

Out ranged values are successfully blocked.



```
=================================================================
Please choose the level you want( 1-easy  2-medium  3-hard ) : #
Input out of range!Input again please: $
Input out of range!Input again please: 2
You have choose level 2
Game start!
Choose the following options please.
=================================================================
```

Signs are successfully blocked.



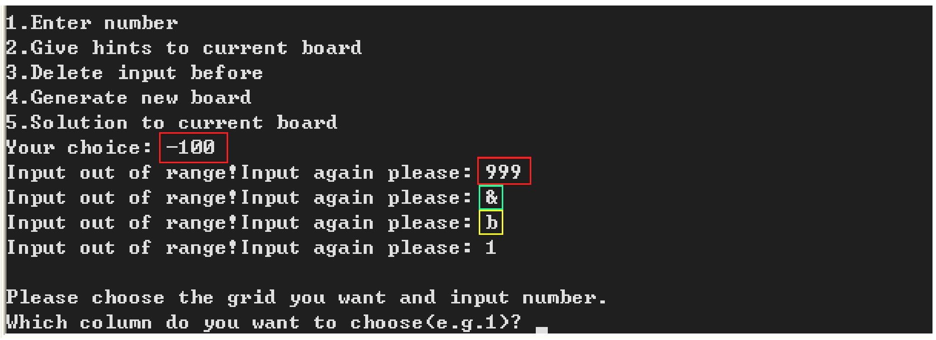
```
=================================================================
Please choose the level you want( 1-easy  2-medium  3-hard ) : a
Input out of range!Input again please: z
Input out of range!Input again please: 3
You have choose level 3
Game start!
Choose the following options please.
=================================================================
```

Characters are successfully blocked.

Choose option

Required input	Required range
Option	1 - 5

Result:



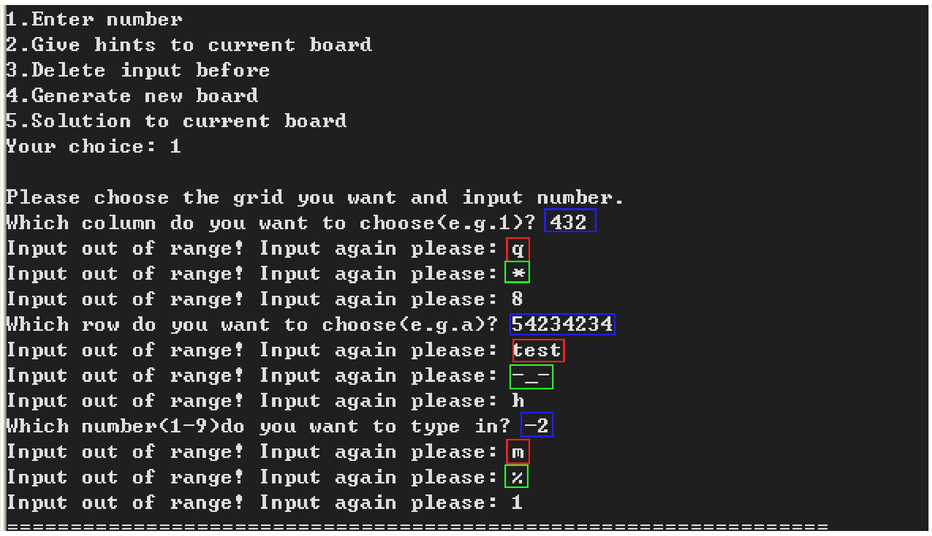
```
1.Enter number
2.Give hints to current board
3.Delete input before
4.Generate new board
5.Solution to current board
Your choice: -100
Input out of range!Input again please: 999
Input out of range!Input again please: &
Input out of range!Input again please: b
Input out of range!Input again please: 1

Please choose the grid you want and input number.
Which column do you want to choose(e.g.1)?
```

Out ranged values, signs and characters are successfully blocked.

Fill in number

Required input	Required range
Row	a - i
Column	1 - 9
number	1 - 9

Result:



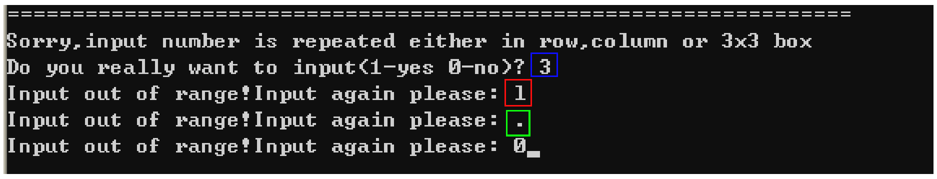
```
1.Enter number
2.Give hints to current board
3.Delete input before
4.Generate new board
5.Solution to current board
Your choice: 1

Please choose the grid you want and input number.
Which column do you want to choose(e.g.1)? 432
Input out of range! Input again please: q
Input out of range! Input again please: *
Input out of range! Input again please: 8
Which row do you want to choose(e.g.a)? 54234234
Input out of range! Input again please: test
Input out of range! Input again please: -_-
Input out of range! Input again please: h
Which number(1-9)do you want to type in? -2
Input out of range! Input again please: m
Input out of range! Input again please: %
Input out of range! Input again please: 1
=================================================================
```

Out ranged values, signs and characters are successfully blocked.

Withdraw wrong input

Required input	Required range
-	0 - 1

Result:



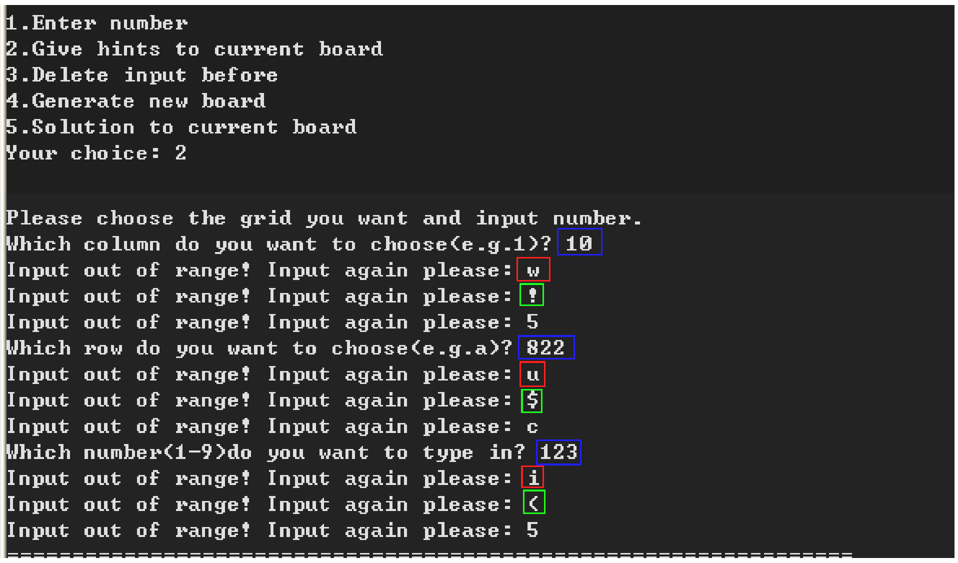
```
=================================================================
Sorry,input number is repeated either in row,column or 3x3 box
Do you really want to input(1-yes 0-no)? 3
Input out of range!Input again please: l
Input out of range!Input again please: .
Input out of range!Input again please: 0
```

Out ranged values, signs and characters are successfully blocked.

Request hints

Required input	Required range
Row	a - i
Column	1 - 9
number	1 - 9

Result:



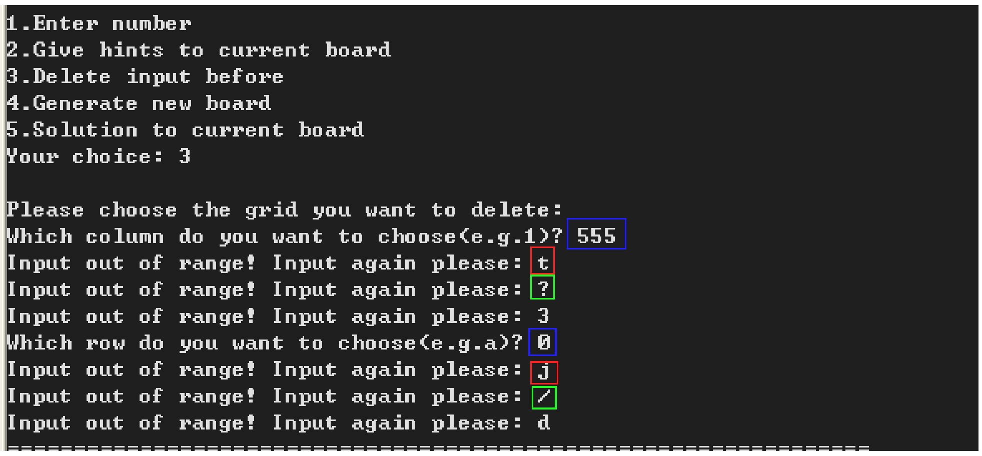
```
1.Enter number
2.Give hints to current board
3.Delete input before
4.Generate new board
5.Solution to current board
Your choice: 2

Please choose the grid you want and input number.
Which column do you want to choose(e.g.1)? 10
Input out of range! Input again please: w
Input out of range! Input again please: !
Input out of range! Input again please: 5
Which row do you want to choose(e.g.a)? 822
Input out of range! Input again please: u
Input out of range! Input again please: $
Input out of range! Input again please: c
Which number(1-9)do you want to type in? 123
Input out of range! Input again please: i
Input out of range! Input again please: <
Input out of range! Input again please: 5
================================================================
```

Out ranged values, signs and characters are successfully blocked.

Delete number

Required input	Required range
Row	a - i
Column	1 - 9
number	1 - 9

Result:



```
1.Enter number
2.Give hints to current board
3.Delete input before
4.Generate new board
5.Solution to current board
Your choice: 3

Please choose the grid you want to delete:
Which column do you want to choose(e.g.1)? 555
Input out of range! Input again please: t
Input out of range! Input again please: ?
Input out of range! Input again please: 3
Which row do you want to choose(e.g.a)? 0
Input out of range! Input again please: j
Input out of range! Input again please: /
Input out of range! Input again please: d
================================================================
```

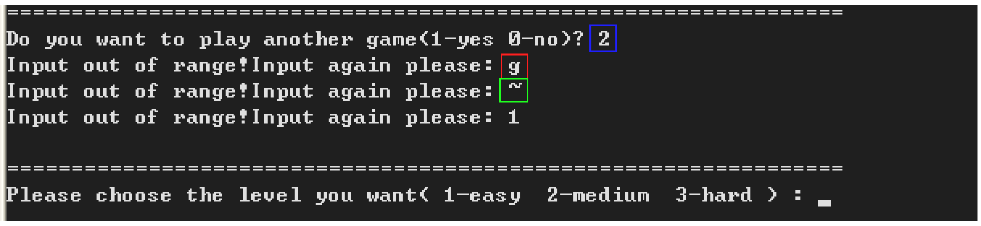
Out ranged values, signs and characters are successfully blocked.

Choose to play another game

Required input	Required range

-	0 - 1

Result:



```
=================================================================
Do you want to play another game(1-yes 0-no)? 2
Input out of range!Input again please: g
Input out of range!Input again please: ~
Input out of range!Input again please: 1

=================================================================
Please choose the level you want( 1-easy  2-medium  3-hard ) : _
```

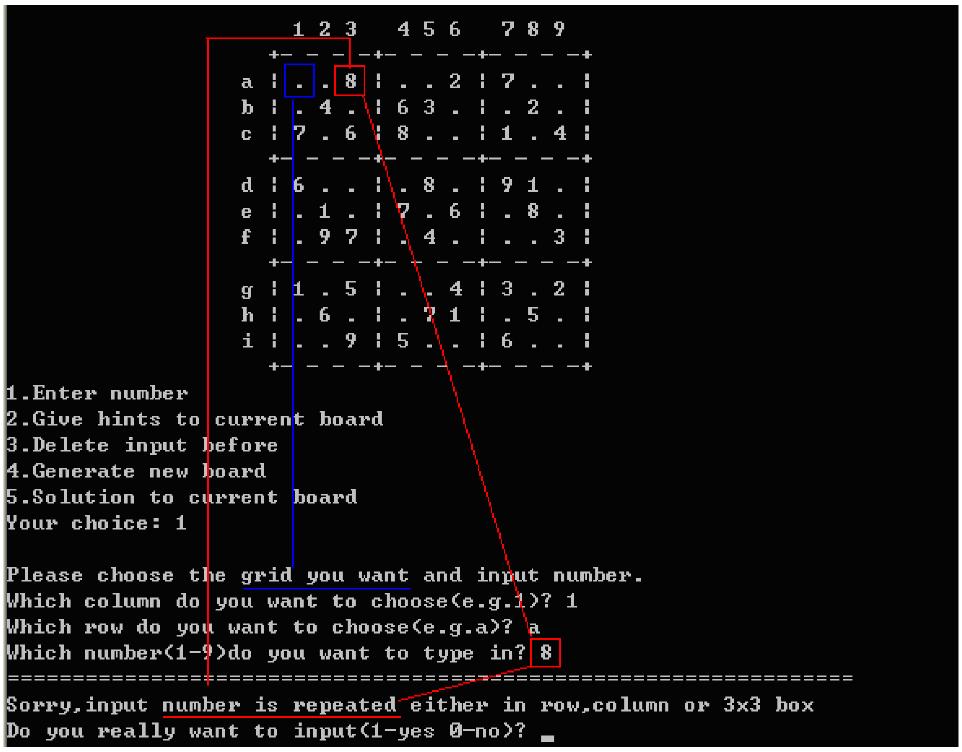
Out ranged values, signs and characters are successfully blocked.

After testing, it proved that my program can ensure the correctness of inputs from users which may affect the procedure of game deeply. Right input is extremely important as wrong on may cause infinite looping of program.

✧ STEP 2:

When a number which is repeated in same row is entered by user:

Result:

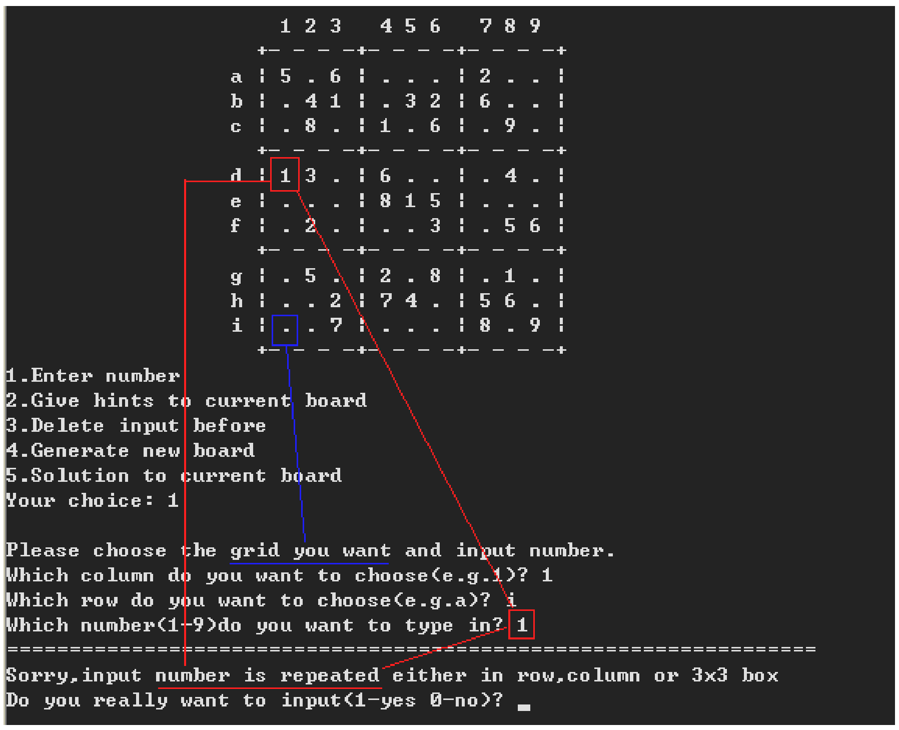


```
                     1 2 3   4 5 6   7 8 9
                   +- - - -+- - - -+- - - -+
                 a | . . 8 | . . 2 | 7 . . |
                 b | . 4 . | 6 3 . | . 2 . |
                 c | 7 . 6 | 8 . . | 1 . 4 |
                   +- - - -+- - - -+- - - -+
                 d | 6 . . | . 8 . | 9 1 . |
                 e | . 1 . | 7 . 6 | . 8 . |
                 f | . 9 7 | . 4 . | . . 3 |
                   +- - - -+- - - -+- - - -+
                 g | 1 . 5 | . . 4 | 3 . 2 |
                 h | . 6 . | . 7 1 | . 5 . |
                 i | . . 9 | 5 . . | 6 . . |
                   +- - - -+- - - -+- - - -+
1.Enter number
2.Give hints to current board
3.Delete input before
4.Generate new board
5.Solution to current board
Your choice: 1

Please choose the grid you want and input number.
Which column do you want to choose(e.g.1)? 1
Which row do you want to choose(e.g.a)? a
Which number(1-9)do you want to type in? 8
=================================================================
Sorry,input number is repeated either in row,column or 3x3 box
Do you really want to input(1-yes 0-no)? _
```

Grid 1a is being chosen, 8 is to be filled in. However, it is appeared in 3a already. My program detected the problem and worn the user immediately.

And that is true for repetition found in column:

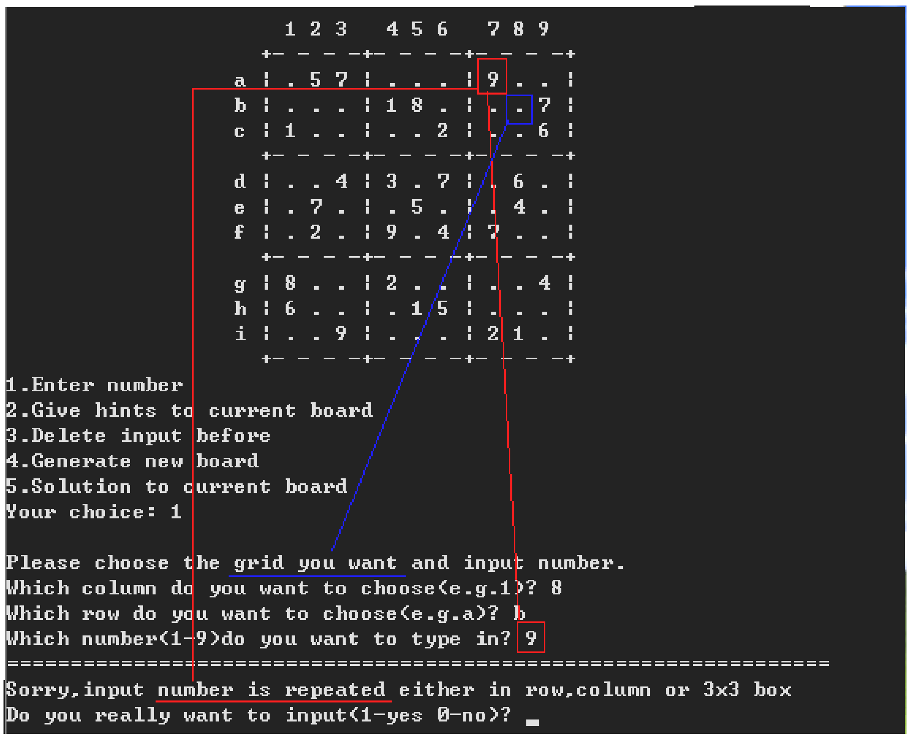
Result:



```
         1 2 3   4 5 6   7 8 9
       +- - - -+- - - -+- - - -+
     a ¦ 5 . 6 ¦ . . . ¦ 2 . . ¦
     b ¦ . 4 1 ¦ . 3 2 ¦ 6 . . ¦
     c ¦ . 8 . ¦ 1 . 6 ¦ . 9 . ¦
       +- - - -+- - - -+- - - -+
     d ¦ 1 3 . ¦ 6 . . ¦ . 4 . ¦
     e ¦ . . . ¦ 8 1 5 ¦ . . . ¦
     f ¦ . 2 . ¦ . . 3 ¦ . 5 6 ¦
       +- - - -+- - - -+- - - -+
     g ¦ . 5 . ¦ 2 . 8 ¦ . 1 . ¦
     h ¦ . . 2 ¦ 7 4 . ¦ 5 6 . ¦
     i ¦ . . 7 ¦ . . . ¦ 8 . 9 ¦
       +- - - -+- - - -+- - - -+
1.Enter number
2.Give hints to current board
3.Delete input before
4.Generate new board
5.Solution to current board
Your choice: 1

Please choose the grid you want and input number.
Which column do you want to choose(e.g.1)? 1
Which row do you want to choose(e.g.a)? i
Which number(1-9)do you want to type in? 1
=================================================================
Sorry,input number is repeated either in row,column or 3x3 box
Do you really want to input(1-yes 0-no)? _
```

It is also true for repetition found in 3x3 boxes:

Result:



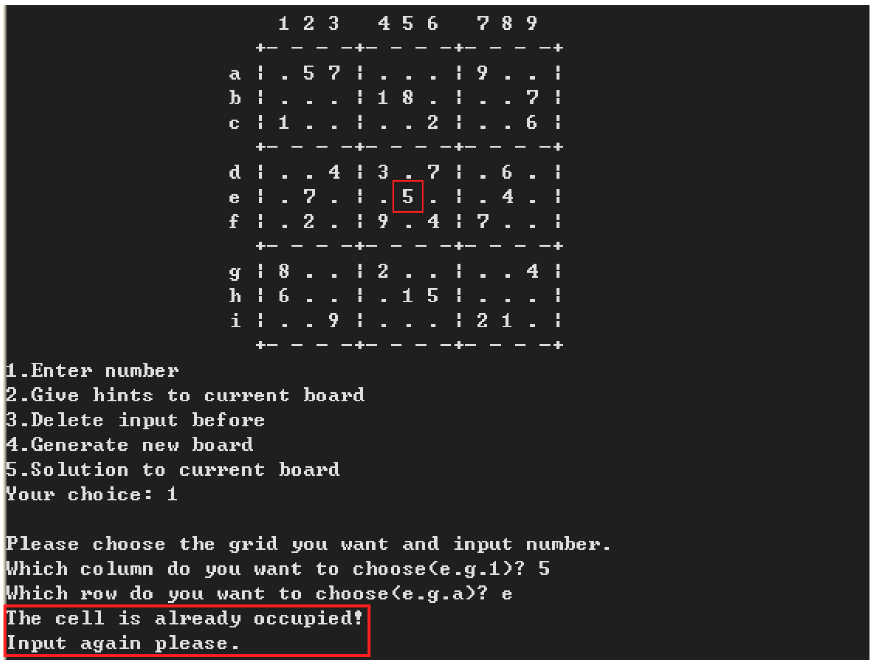
```
         1 2 3   4 5 6   7 8 9
       +- - - -+- - - -+- - - -+
     a ¦ . 5 7 ¦ . . . ¦ 9 . . ¦
     b ¦ . . . ¦ 1 8 . ¦ . . 7 ¦
     c ¦ 1 . . ¦ . . 2 ¦ . . 6 ¦
       +- - - -+- - - -+- - - -+
     d ¦ . . 4 ¦ 3 . 7 ¦ . 6 . ¦
     e ¦ . 7 . ¦ . 5 . ¦ . 4 . ¦
     f ¦ . 2 . ¦ 9 . 4 ¦ 7 . . ¦
       +- - - -+- - - -+- - - -+
     g ¦ 8 . . ¦ 2 . . ¦ . . 4 ¦
     h ¦ 6 . . ¦ . 1 5 ¦ . . . ¦
     i ¦ . . 9 ¦ . . . ¦ 2 1 . ¦
       +- - - -+- - - -+- - - -+
1.Enter number
2.Give hints to current board
3.Delete input before
4.Generate new board
5.Solution to current board
Your choice: 1

Please choose the grid you want and input number.
Which column do you want to choose(e.g.1)? 8
Which row do you want to choose(e.g.a)? b
Which number(1-9)do you want to type in? 9
=================================================================
Sorry,input number is repeated either in row,column or 3x3 box
Do you really want to input(1-yes 0-no)? _
```

There is no 9 found in same row or column, but just same 3x3 boxes.

This function for checking allowance of input can function normally whenever repetitions are found either in same row, column or 3x3 boxes.

If user choose grid with content already:

Result:



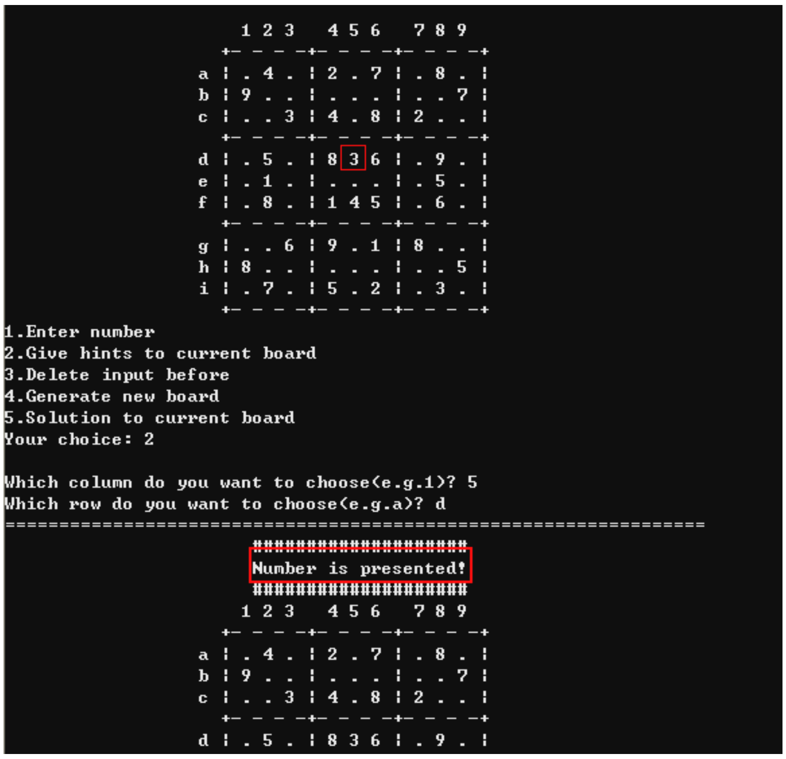
```
             1 2 3   4 5 6   7 8 9
           +- - - -+- - - -+- - - -+
         a ¦ . 5 7 ¦ . . . ¦ 9 . . ¦
         b ¦ . . . ¦ 1 8 . ¦ . . 7 ¦
         c ¦ 1 . . ¦ . . 2 ¦ . . 6 ¦
           +- - - -+- - - -+- - - -+
         d ¦ . . 4 ¦ 3 . 7 ¦ . 6 . ¦
         e ¦ . 7 . ¦ . 5 . ¦ . 4 . ¦
         f ¦ . 2 . ¦ 9 . 4 ¦ 7 . . ¦
           +- - - -+- - - -+- - - -+
         g ¦ 8 . . ¦ 2 . . ¦ . . 4 ¦
         h ¦ 6 . . ¦ . 1 5 ¦ . . . ¦
         i ¦ . . 9 ¦ . . . ¦ 2 1 . ¦
           +- - - -+- - - -+- - - -+
1.Enter number
2.Give hints to current board
3.Delete input before
4.Generate new board
5.Solution to current board
Your choice: 1

Please choose the grid you want and input number.
Which column do you want to choose(e.g.1)? 5
Which row do you want to choose(e.g.a)? e
The cell is already occupied!
Input again please.
```

Input is forbidden unless it is both correct and valid input.

✧ STEP 3:

For function give hint, if user choose grid with number presented already:

Result:

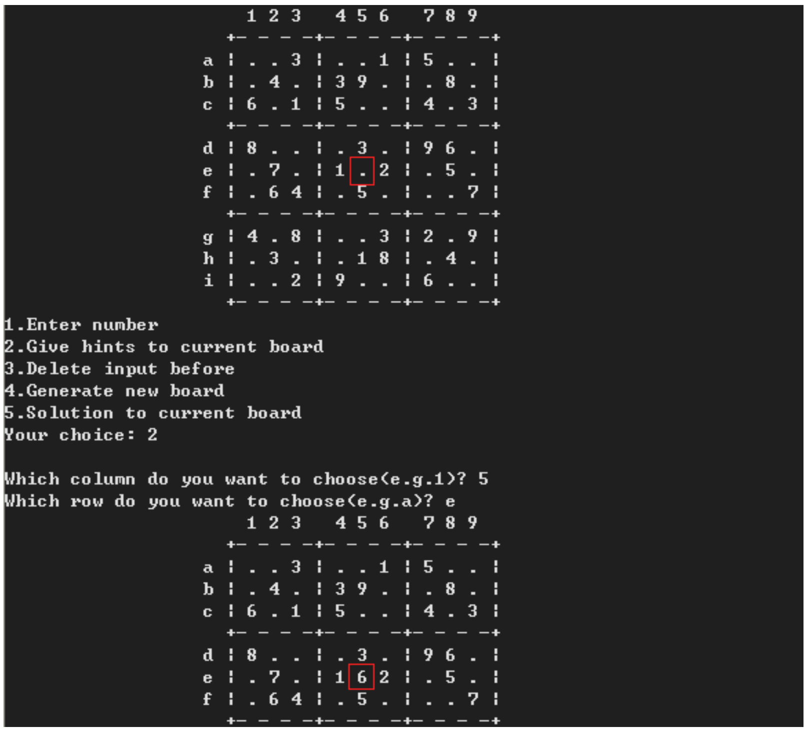


```
             1 2 3   4 5 6   7 8 9
           +- - - -+- - - -+- - - -+
         a ¦ . 4 . ¦ 2 . 7 ¦ . 8 . ¦
         b ¦ 9 . . ¦ . . . ¦ . . 7 ¦
         c ¦ . . 3 ¦ 4 . 8 ¦ 2 . . ¦
           +- - - -+- - - -+- - - -+
         d ¦ . 5 . ¦ 8 3 6 ¦ . 9 . ¦
         e ¦ . 1 . ¦ . . . ¦ . 5 . ¦
         f ¦ . 8 . ¦ 1 4 5 ¦ . 6 . ¦
           +- - - -+- - - -+- - - -+
         g ¦ . . 6 ¦ 9 . 1 ¦ 8 . . ¦
         h ¦ 8 . . ¦ . . . ¦ . . 5 ¦
         i ¦ . 7 . ¦ 5 . 2 ¦ . 3 . ¦
           +- - - -+- - - -+- - - -+
1.Enter number
2.Give hints to current board
3.Delete input before
4.Generate new board
5.Solution to current board
Your choice: 2

Which column do you want to choose(e.g.1)? 5
Which row do you want to choose(e.g.a)? d
=================================================================
                      ####################
                      Number is presented!
                      ####################
             1 2 3   4 5 6   7 8 9
           +- - - -+- - - -+- - - -+
         a ¦ . 4 . ¦ 2 . 7 ¦ . 8 . ¦
         b ¦ 9 . . ¦ . . . ¦ . . 7 ¦
         c ¦ . . 3 ¦ 4 . 8 ¦ 2 . . ¦
           +- - - -+- - - -+- - - -+
         d ¦ . 5 . ¦ 8 3 6 ¦ . 9 . ¦
```

Warning message came out, the step had no effect.

If the grid did not contain anything:

Result:



```
              1 2 3   4 5 6   7 8 9
            +- - - -+- - - -+- - - -+
          a ¦ . . 3 ¦ . . 1 ¦ 5 . . ¦
          b ¦ . 4 . ¦ 3 9 . ¦ . 8 . ¦
          c ¦ 6 . 1 ¦ 5 . . ¦ 4 . 3 ¦
            +- - - -+- - - -+- - - -+
          d ¦ 8 . . ¦ . 3 . ¦ 9 6 . ¦
          e ¦ . 7 . ¦ 1 . 2 ¦ . 5 . ¦
          f ¦ . 6 4 ¦ . 5 . ¦ . . 7 ¦
            +- - - -+- - - -+- - - -+
          g ¦ 4 . 8 ¦ . . 3 ¦ 2 . 9 ¦
          h ¦ . 3 . ¦ . 1 8 ¦ . 4 . ¦
          i ¦ . . 2 ¦ 9 . . ¦ 6 . . ¦
            +- - - -+- - - -+- - - -+
1.Enter number
2.Give hints to current board
3.Delete input before
4.Generate new board
5.Solution to current board
Your choice: 2

Which column do you want to choose(e.g.1)? 5
Which row do you want to choose(e.g.a)? e
              1 2 3   4 5 6   7 8 9
            +- - - -+- - - -+- - - -+
          a ¦ . . 3 ¦ . . 1 ¦ 5 . . ¦
          b ¦ . 4 . ¦ 3 9 . ¦ . 8 . ¦
          c ¦ 6 . 1 ¦ 5 . . ¦ 4 . 3 ¦
            +- - - -+- - - -+- - - -+
          d ¦ 8 . . ¦ . 3 . ¦ 9 6 . ¦
          e ¦ . 7 . ¦ 1 6 2 ¦ . 5 . ¦
          f ¦ . 6 4 ¦ . 5 . ¦ . . 7 ¦
            +- - - -+- - - -+- - - -+
```

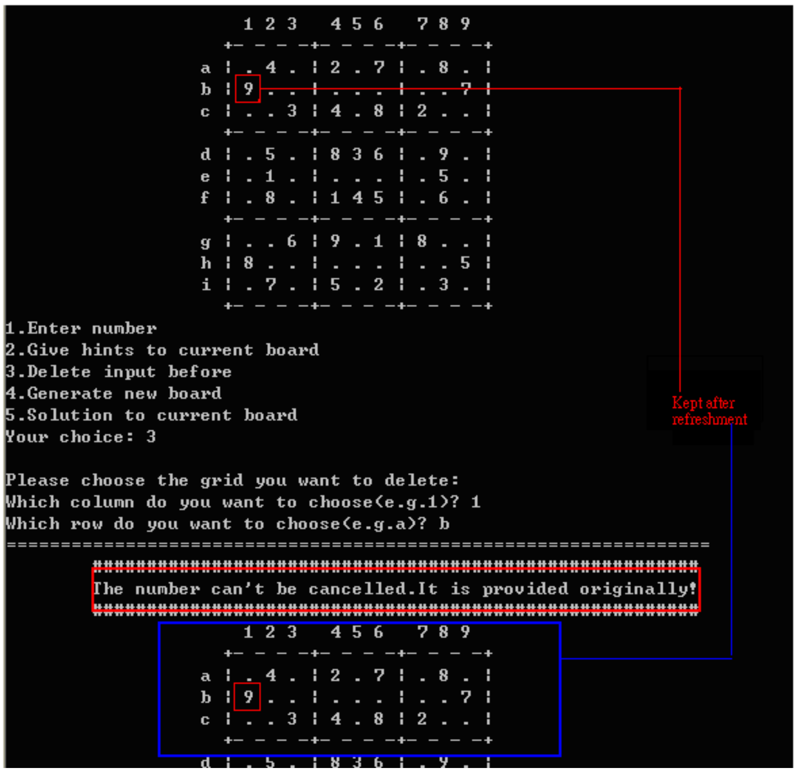
After refreshment, the correct answer read from solution I prepared was appeared on board.

This function can produce anticipate result as my expectation. Thus it is successfully implanted.

✧ STEP 4:

For function delete number, if user choose grid with number given at the beginning:

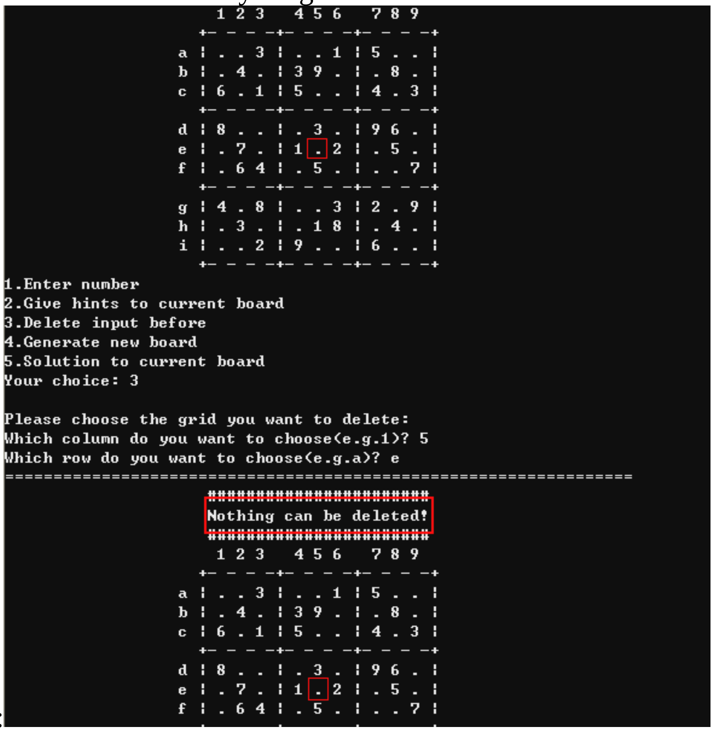
Result:



```
              1 2 3   4 5 6   7 8 9
            +- - - -+- - - -+- - - -+
          a ¦ . 4 . ¦ 2 . 7 ¦ . 8 . ¦
          b ¦ 9 . . ¦ . . . ¦ . . 7 ¦
          c ¦ . . 3 ¦ 4 . 8 ¦ 2 . . ¦
            +- - - -+- - - -+- - - -+
          d ¦ . 5 . ¦ 8 3 6 ¦ . 9 . ¦
          e ¦ . 1 . ¦ . . . ¦ . 5 . ¦
          f ¦ . 8 . ¦ 1 4 5 ¦ . 6 . ¦
            +- - - -+- - - -+- - - -+
          g ¦ . . 6 ¦ 9 . 1 ¦ 8 . . ¦
          h ¦ 8 . . ¦ . . . ¦ . . 5 ¦
          i ¦ . 7 . ¦ 5 . 2 ¦ . 3 . ¦
            +- - - -+- - - -+- - - -+
1.Enter number
2.Give hints to current board
3.Delete input before
4.Generate new board
5.Solution to current board
Your choice: 3

Please choose the grid you want to delete:
Which column do you want to choose(e.g.1)? 1
Which row do you want to choose(e.g.a)? b
=================================================================
        ########################################################
        The number can't be cancelled.It is provided originally!
        ########################################################
              1 2 3   4 5 6   7 8 9
            +- - - -+- - - -+- - - -+
          a ¦ . 4 . ¦ 2 . 7 ¦ . 8 . ¦
          b ¦ 9 . . ¦ . . . ¦ . . 7 ¦
          c ¦ . . 3 ¦ 4 . 8 ¦ 2 . . ¦
            +- - - -+- - - -+- - - -+
          d ¦ . 5 . ¦ 8 3 6 ¦ . 9 . ¦
```

The number is unable to be deleted.

If the grid did not contain anything:



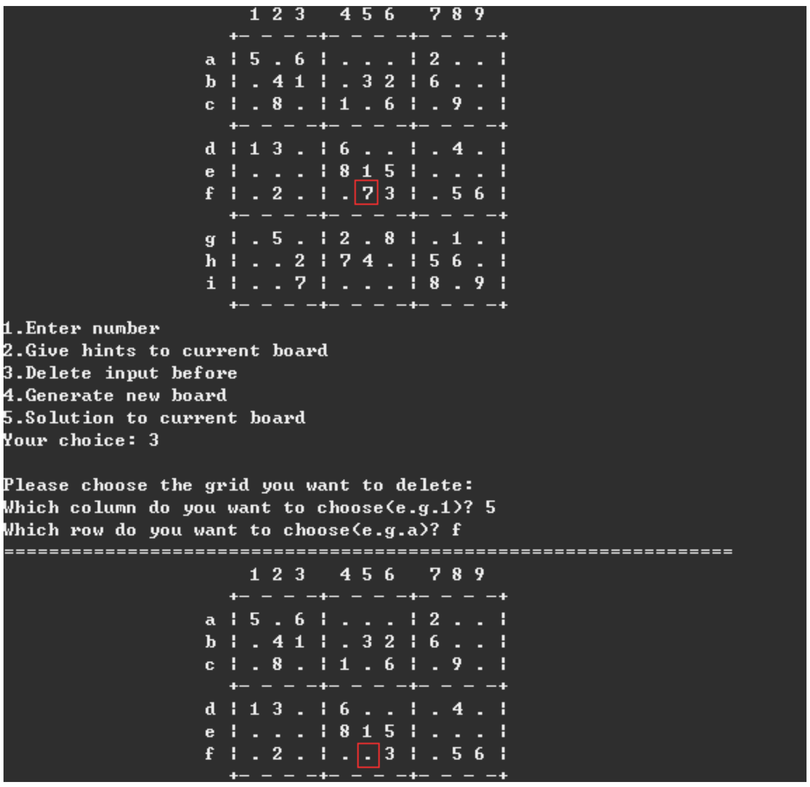
```
       1 2 3   4 5 6   7 8 9
     +- - - -+- - - -+- - - -+
   a ¦ . . 3 ¦ . . 1 ¦ 5 . . ¦
   b ¦ . 4 . ¦ 3 9 . ¦ . 8 . ¦
   c ¦ 6 . 1 ¦ 5 . . ¦ 4 . 3 ¦
     +- - - -+- - - -+- - - -+
   d ¦ 8 . . ¦ . 3 . ¦ 9 6 . ¦
   e ¦ . 7 . ¦ 1 . 2 ¦ . 5 . ¦
   f ¦ . 6 4 ¦ . 5 . ¦ . . 7 ¦
     +- - - -+- - - -+- - - -+
   g ¦ 4 . 8 ¦ . . 3 ¦ 2 . 9 ¦
   h ¦ . 3 . ¦ . 1 8 ¦ . 4 . ¦
   i ¦ . . 2 ¦ 9 . . ¦ 6 . . ¦
     +- - - -+- - - -+- - - -+
1.Enter number
2.Give hints to current board
3.Delete input before
4.Generate new board
5.Solution to current board
Your choice: 3

Please choose the grid you want to delete:
Which column do you want to choose(e.g.1)? 5
Which row do you want to choose(e.g.a)? e
=================================================================
                    #######################
                    Nothing can be deleted!
                    #######################
       1 2 3   4 5 6   7 8 9
     +- - - -+- - - -+- - - -+
   a ¦ . . 3 ¦ . . 1 ¦ 5 . . ¦
   b ¦ . 4 . ¦ 3 9 . ¦ . 8 . ¦
   c ¦ 6 . 1 ¦ 5 . . ¦ 4 . 3 ¦
     +- - - -+- - - -+- - - -+
   d ¦ 8 . . ¦ . 3 . ¦ 9 6 . ¦
   e ¦ . 7 . ¦ 1 . 2 ¦ . 5 . ¦
   f ¦ . 6 4 ¦ . 5 . ¦ . . 7 ¦
```

:

Warning message came out, the step had no effect.

But if the number is inputted by users:

Result:



```
       1 2 3   4 5 6   7 8 9
     +- - - -+- - - -+- - - -+
   a ¦ 5 . 6 ¦ . . . ¦ 2 . . ¦
   b ¦ . 4 1 ¦ . 3 2 ¦ 6 . . ¦
   c ¦ . 8 . ¦ 1 . 6 ¦ . 9 . ¦
     +- - - -+- - - -+- - - -+
   d ¦ 1 3 . ¦ 6 . . ¦ . 4 . ¦
   e ¦ . . . ¦ 8 1 5 ¦ . . . ¦
   f ¦ . 2 . ¦ . 7 3 ¦ . 5 6 ¦
     +- - - -+- - - -+- - - -+
   g ¦ . 5 . ¦ 2 . 8 ¦ . 1 . ¦
   h ¦ . . 2 ¦ 7 4 . ¦ 5 6 . ¦
   i ¦ . . 7 ¦ . . . ¦ 8 . 9 ¦
     +- - - -+- - - -+- - - -+
1.Enter number
2.Give hints to current board
3.Delete input before
4.Generate new board
5.Solution to current board
Your choice: 3

Please choose the grid you want to delete:
Which column do you want to choose(e.g.1)? 5
Which row do you want to choose(e.g.a)? f
=================================================================
       1 2 3   4 5 6   7 8 9
     +- - - -+- - - -+- - - -+
   a ¦ 5 . 6 ¦ . . . ¦ 2 . . ¦
   b ¦ . 4 1 ¦ . 3 2 ¦ 6 . . ¦
   c ¦ . 8 . ¦ 1 . 6 ¦ . 9 . ¦
     +- - - -+- - - -+- - - -+
   d ¦ 1 3 . ¦ 6 . . ¦ . 4 . ¦
   e ¦ . . . ¦ 8 1 5 ¦ . . . ¦
   f ¦ . 2 . ¦ . . 3 ¦ . 5 6 ¦
     +- - - -+- - - -+- - - -+
```

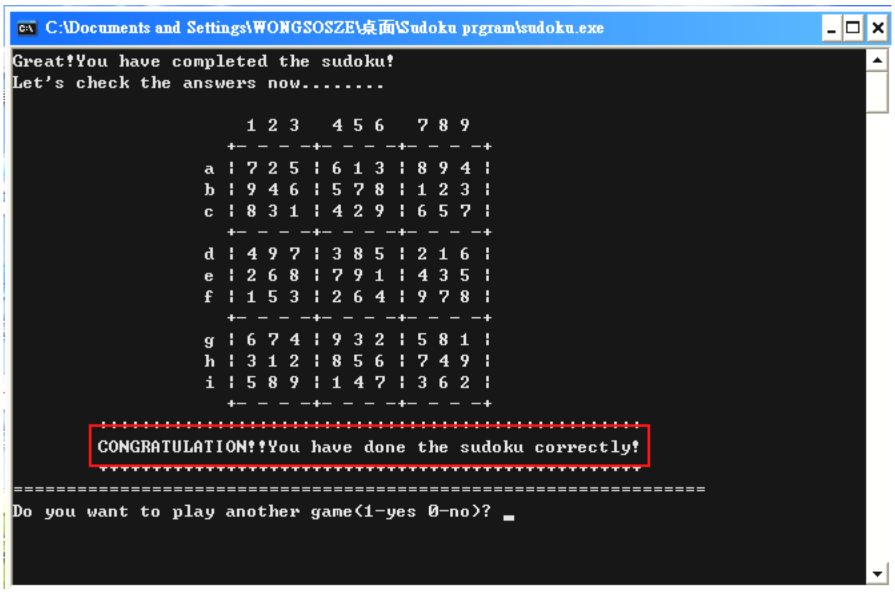
Number is successfully being deleted.

This function can produce anticipate result as my expectation. Thus it is successfully implanted.

✧ STEP 5:

When the final solution given by users got no mistakes:

Result:

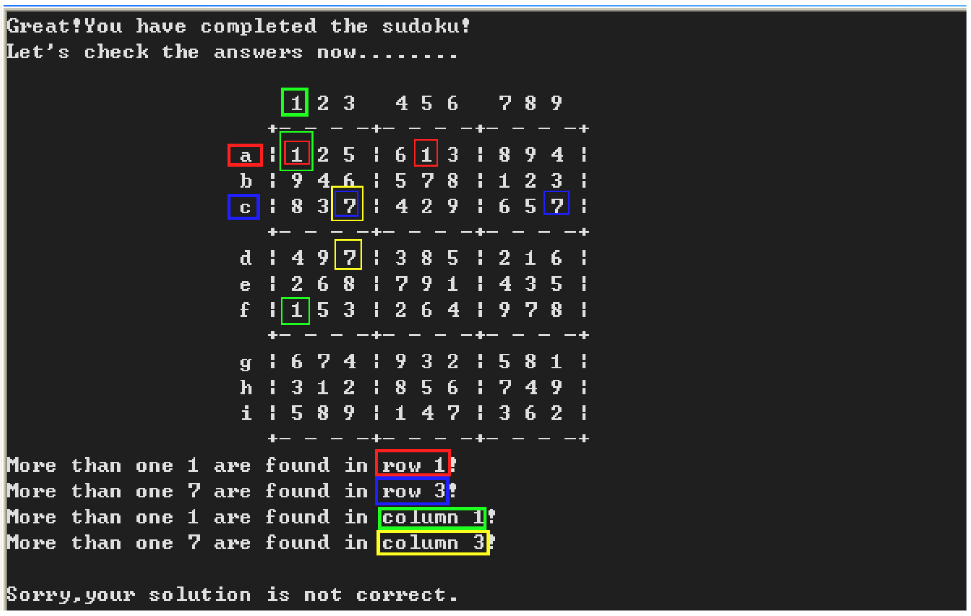

C:\Documents and Settings\WONGSOSZE\桌面\Sudoku prgram\sudoku.exe

```
Great!You have completed the sudoku!
Let's check the answers now........

                      1 2 3   4 5 6   7 8 9
                    +- - - -+- - - -+- - - -+
                  a ¦ 7 2 5 ¦ 6 1 3 ¦ 8 9 4 ¦
                  b ¦ 9 4 6 ¦ 5 7 8 ¦ 1 2 3 ¦
                  c ¦ 8 3 1 ¦ 4 2 9 ¦ 6 5 7 ¦
                    +- - - -+- - - -+- - - -+
                  d ¦ 4 9 7 ¦ 3 8 5 ¦ 2 1 6 ¦
                  e ¦ 2 6 8 ¦ 7 9 1 ¦ 4 3 5 ¦
                  f ¦ 1 5 3 ¦ 2 6 4 ¦ 9 7 8 ¦
                    +- - - -+- - - -+- - - -+
                  g ¦ 6 7 4 ¦ 9 3 2 ¦ 5 8 1 ¦
                  h ¦ 3 1 2 ¦ 8 5 6 ¦ 7 4 9 ¦
                  i ¦ 5 8 9 ¦ 1 4 7 ¦ 3 6 2 ¦
                    +- - - -+- - - -+- - - -+
        !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
        CONGRATULATION!!You have done the sudoku correctly!
        ***************************************************
=================================================================
Do you want to play another game(1-yes 0-no)? _
```

My program realizes correct solution and congratulation message is printed out.

When the final solution given by users contain mistakes on some rows and columns:

Result:

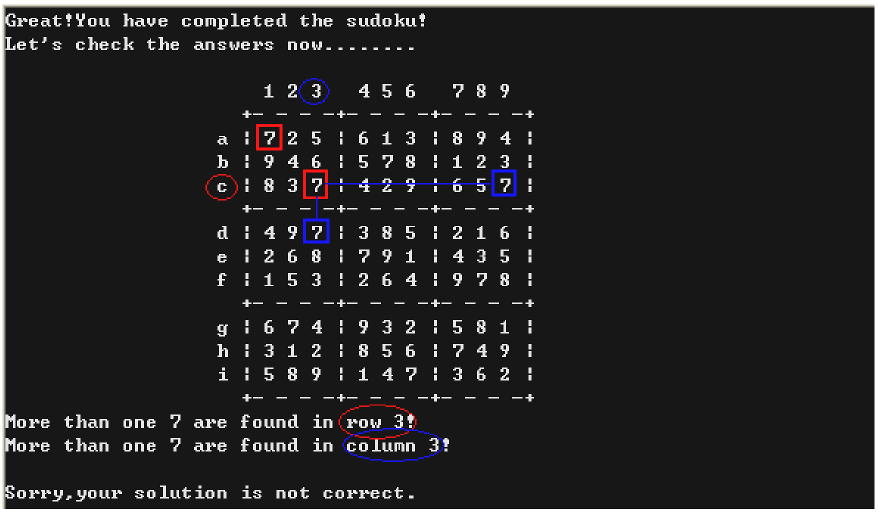


```
Great!You have completed the sudoku!
Let's check the answers now........

                      1 2 3   4 5 6   7 8 9
                    +- - - -+- - - -+- - - -+
                  a ¦ 1 2 5 ¦ 6 1 3 ¦ 8 9 4 ¦
                  b ¦ 9 4 6 ¦ 5 7 8 ¦ 1 2 3 ¦
                  c ¦ 8 3 7 ¦ 4 2 9 ¦ 6 5 7 ¦
                    +- - - -+- - - -+- - - -+
                  d ¦ 4 9 7 ¦ 3 8 5 ¦ 2 1 6 ¦
                  e ¦ 2 6 8 ¦ 7 9 1 ¦ 4 3 5 ¦
                  f ¦ 1 5 3 ¦ 2 6 4 ¦ 9 7 8 ¦
                    +- - - -+- - - -+- - - -+
                  g ¦ 6 7 4 ¦ 9 3 2 ¦ 5 8 1 ¦
                  h ¦ 3 1 2 ¦ 8 5 6 ¦ 7 4 9 ¦
                  i ¦ 5 8 9 ¦ 1 4 7 ¦ 3 6 2 ¦
                    +- - - -+- - - -+- - - -+
More than one 1 are found in row 1!
More than one 7 are found in row 3!
More than one 1 are found in column 1!
More than one 7 are found in column 3!

Sorry,your solution is not correct.
```

The solution is not correct. My program correctly detected all the mistakes.

When the final solution given by users contain mistake on same box:

Result:



```
Great!You have completed the sudoku!
Let's check the answers now........

                    1 2 3   4 5 6   7 8 9
                  +- - - -+- - - -+- - - -+
                a | 7 2 5 | 6 1 3 | 8 9 4 |
                b | 9 4 6 | 5 7 8 | 1 2 3 |
                c | 8 3 7 | 4 2 9 | 6 5 7 |
                  +- - - -+- - - -+- - - -+
                d | 4 9 7 | 3 8 5 | 2 1 6 |
                e | 2 6 8 | 7 9 1 | 4 3 5 |
                f | 1 5 3 | 2 6 4 | 9 7 8 |
                  +- - - -+- - - -+- - - -+
                g | 6 7 4 | 9 3 2 | 5 8 1 |
                h | 3 1 2 | 8 5 6 | 7 4 9 |
                i | 5 8 9 | 1 4 7 | 3 6 2 |
                  +- - - -+- - - -+- - - -+
More than one 7 are found in row 3!
More than one 7 are found in column 3!

Sorry,your solution is not correct.
```

Although I have not written codes for checking 3x3 boxes at the end, my program still can nose out mistakes in same box by help of same number in same row and column.

This function can produce anticipate result as my expectation. Despite the user's solution is not the same as my solution. I believe mistake can also be detected out. But the problem is I can't find alternate set of solution that can fit those games I prepared.

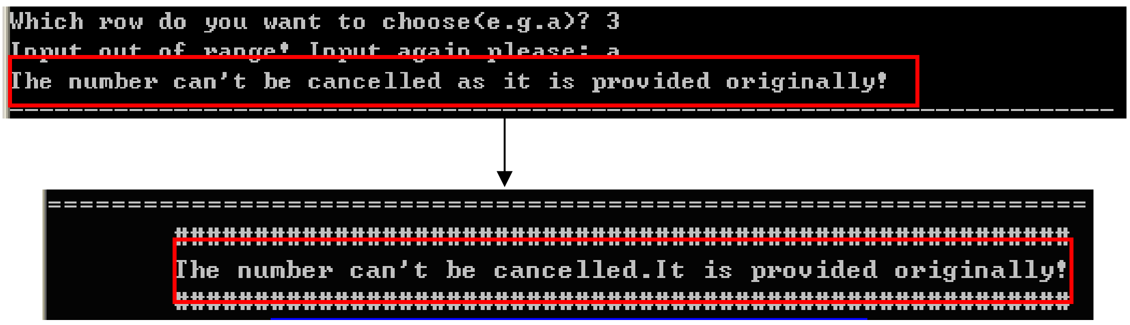
➢ Testing (by friends):

Besides testing the program by myself, I have also invited my sister and friends I met on the Internet to play my game in order to examine the user-friendliness of my program,, their feedbacks are collected and listed below. Some changes are made instantly:

✧ Some important statements are pushed upward when the board refresh and not eye catching. So they feel difficult to follow them.

Solution: I try my best to shorten those important messages and embedded them into lines constructed using symbols so that they can be discovered easily.

E.g.



```
Which row do you want to choose(e.g.a)? 3
Input out of range! Input again please: a
The number can't be cancelled as it is provided originally!
------------------------------------------------------------------------
```

```
====================================================================
        ##########################################################
        The number can't be cancelled.It is provided originally!
        ##########################################################
```

✧ The program is flooded with messages as the program keep on refreshing, it is really an undesirable thing.

Solution: I added in a function called **system("CLS")** at certain place. Now the messages generated before will be cleared after user has inputted number to the board.

➢ Evaluation:

After testing every important part of my program, the results are found to be suiting my expectation. There isn't any serious mistake. So I can guarantee that my program can run normally.

My testing plan may not be perfect. But I have already emended some minor mistakes during the process. For example, numbers given by program can be deleted at first, I discovered this accidentally. I regard it as a quite serious problem because all numbers can be removed, leaving behind is a 9 x 9 grid without anything! Later I solved the problem to ensure given tips wouldn't be removed which may make players get stuck.

I think testing is yet another important part in constructing a program because a program with lots of mistakes is useless. I hope my program is error-free after this process.

Comments from my friends and sister are quite positive. They help me a lot. If time is allowed, I would have asked more people to try my program.

Conclusion

➢ Summary:

To sum up, the program is successfully implemented. It can meet the basic requirements. All functions can run normally. As I have just get involved in C programming language for about one years, I feel satisfy with my work.

Throughout the whole working process, I acquired lots of knowledge. It is fantastic that different functions grouping together can produce a 'magic'. I admire more about the beauty of function. I have also learnt important debugging skills to increase my working speed like using // or /**/. But of course I know the best way is to avoid errors. My logical thinking ability has also been ameliorated since I have encountered lots of problems requiring good logical thinking skill. So I will not afraid of facing logic challenge any more.

To me, it is really a big challenge as I have never met such large scaled projects before. At the very beginning, I felt quite helpless. But after receiving ideas from teacher and classmates, I can eventually solve the problem. So I feel pride of myself as I can put it through. It has built up my confidence not only in doing CIT project, but also jobs of other subjects. Although it took me a few days which cause reduction in my time doing revision, it is worth devoting.

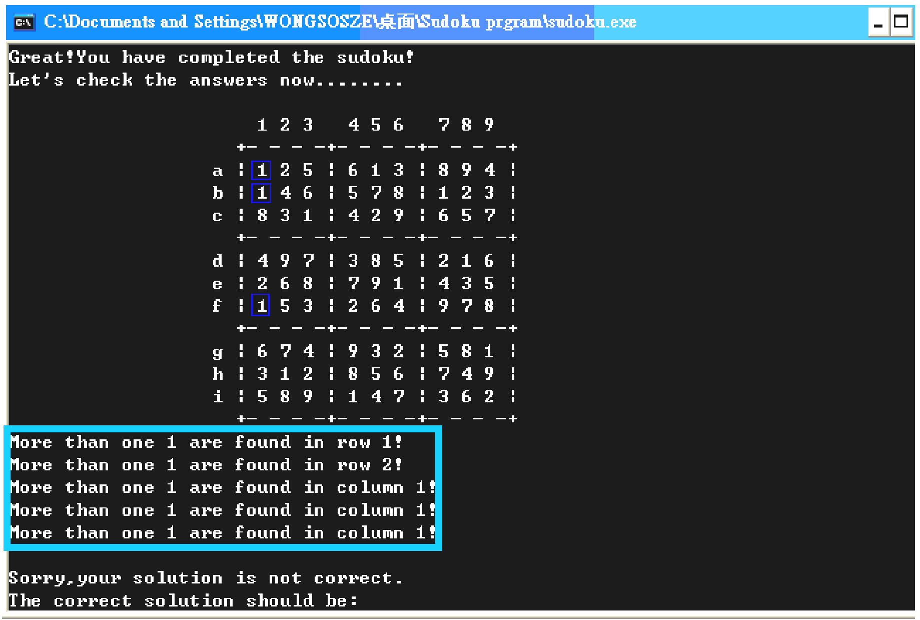
If you ask me whether I had made any wrong choices in doing this project, I would answer 'no' undoubtedly. I don't think I had chosen the wrong language, wrong tools or wrong methods. I will only think that whenever there is a way, there is a will. You will finally achieve what you want if you keep going with the method. As I have to plan the time spending on doing this work, I learnt better self- learning and time management skills. So now the only thing makes me feel regret is I start too late in doing this work. Therefore I will be aware of this starting from now on and it will be the same in my future.

Lastly, I must say thank you to my CIT subject's teacher. Without his guidance and reminders, I wouldn't have finished the project so smoothly.

➢ Problems unsolved:

As I am a novice of programming language, it is tough task for me to strike for a perfect program. So there are some problems left behind.

First of all, when the solution of user do not match with either rules or my solution, my program will list out where repetitions are found as the following:



```
C:\Documents and Settings\WONGSOSZE\桌面\Sudoku prgram\sudoku.exe

Great!You have completed the sudoku!
Let's check the answers now........

           1 2 3   4 5 6   7 8 9
          +- - - -+- - - -+- - - -+
        a ¦ 1 2 5 ¦ 6 1 3 ¦ 8 9 4 ¦
        b ¦ 1 4 6 ¦ 5 7 8 ¦ 1 2 3 ¦
        c ¦ 8 3 1 ¦ 4 2 9 ¦ 6 5 7 ¦
          +- - - -+- - - -+- - - -+
        d ¦ 4 9 7 ¦ 3 8 5 ¦ 2 1 6 ¦
        e ¦ 2 6 8 ¦ 7 9 1 ¦ 4 3 5 ¦
        f ¦ 1 5 3 ¦ 2 6 4 ¦ 9 7 8 ¦
          +- - - -+- - - -+- - - -+
        g ¦ 6 7 4 ¦ 9 3 2 ¦ 5 8 1 ¦
        h ¦ 3 1 2 ¦ 8 5 6 ¦ 7 4 9 ¦
        i ¦ 5 8 9 ¦ 1 4 7 ¦ 3 6 2 ¦
          +- - - -+- - - -+- - - -+
More than one 1 are found in row 1!
More than one 1 are found in row 2!
More than one 1 are found in column 1!
More than one 1 are found in column 1!
More than one 1 are found in column 1!

Sorry,your solution is not correct.
The correct solution should be:
```

But if there is more than two of a same number found in column or row, the warning message will appear more than one times. I have tried to edit the code, but I failed and the result is even worse. So it is left unsolved.

Also, I intended to check the user's solution against rules. So I tried to write the check part check row, check column and check 3X3box at first respectively. Later it is found that check against solution is another choice. I did not adopt this checking method so as to avoid wrong judgment of alternate solutions. You may feel odd that check 3X3box is missed in my program. But I discovered that if there is any repetitions found in boxes, check row and check column will certainly point out the mistakes. Therefore check 3X3box can be omitted.

Besides, function for checking boxes after finishing the game is much more difficult to write compared with that in instant check part. As a result, I give up in composing it.

➢ Possible Improvements

Although my program is successfully created, further improvements are possible to take place.

The following are possible further improvements in connection with my program:

- The user interface can be amended into more playable one such as graphical user interface. GUI is both more user-friendly and attractive. Users always prefer buttons clicking rather than typing because of its convenience. My teacher has advised me implanting my program into another interface using a library called ‘Curses’. The interface is definitely better. However, due to my poor time- management, I suffered from a failure.

- More text files comprising of games can be added to further increase the interesting level of my program. User thus possesses more choices.
- Some extra functions like time counting can be added in. Users can compare the time they use to finishing a game. It is even more thinkable if comments can be given according to the time they take.
- The method of generating a game can also be changed. Instead of read in a file, the program can generate by it selves. Thus time in creating text files is saved and variation of game is greatly increased.

- The input method can be further improved. Keys that required can set to concentrate on a corner to reduce the movement of user’s hand. For e.g. column keys can be changed into q,w,e,r,t,y,u,i and o. This make input become more convenient

Documentation

Helps from reference books and websites are key elements leading me to finish the task. I Had read and surf the following books and websites respectively:

➢ Books:

Name	Author	Uses
新 C 學習繪本	ANK Co., Ltd	Learn the basic concepts of writing program like structure of functions.
C programming	Raymond W.N. CHAN	Gone through some useful program which gives me inspirations.
Programming in C	Stephen Kochan	Learn methods of debugging.

➢ Websites:

URL	Uses
http://www.is.cityu.edu.hk/staff/ismikeho/is4234/is4234.htm	View some basic concepts
http://cplusplus.com/	Download some programs and try to get inspirations from it.