

EE2310 Introduction to Programming - Assignment 2

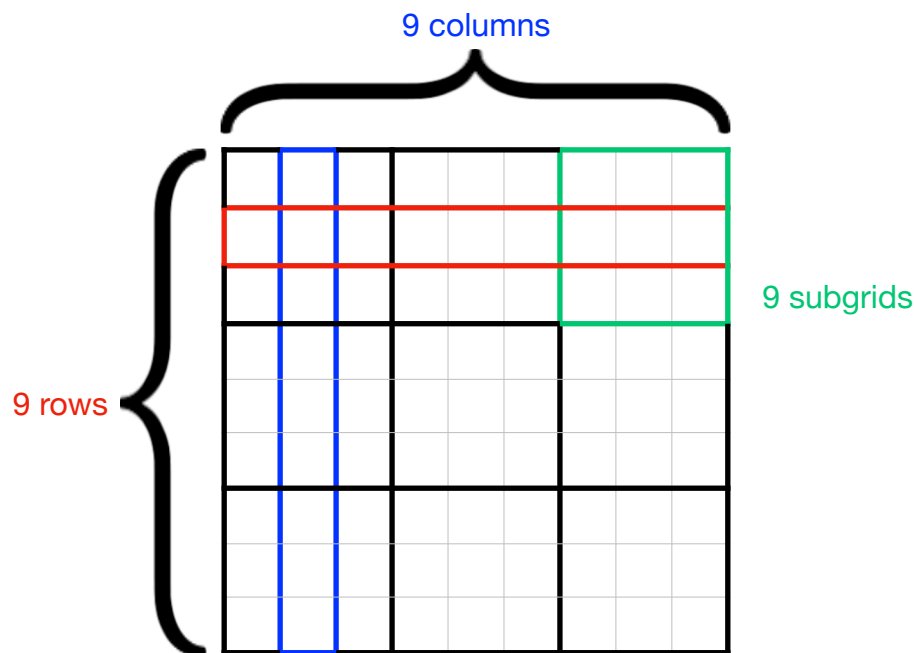
Due: **December 31, 2017 (11:59pm)**

Instructions

1. Submit your code on [NTHU online judge](#) and [iLMS system](#).
2. Name your source code as your student ID. For example, 103064533.c
3. Late submission will incur 8% penalty per day up to 5 days.

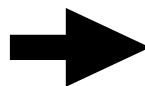
Sudoku

Sudoku, originally called Number Place, is a logic-based, combinatorial number-placement puzzle. The objective is to fill a 9×9 grid with digits so that each column, each row and each 3×3 subgrids (the 3×3 bold edge in the following figure) that compose the grid contain all of the digits from 1 to 9. In other words, there is no repeated number in each column, each row and each subgrid. The puzzle setter provides a partially completed grid, which for a well-posed puzzle has a single solution.



A typical sudoku puzzle:

5	3			7				
6			1	9	5			
	9	8					6	
8				6				3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9



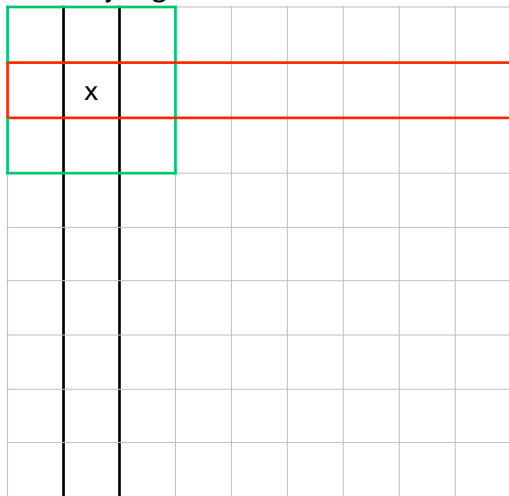
Its solution:

5	3	4	6	7	8	9	1	2
6	7	2	1	9	5	3	4	8
1	9	8	3	4	2	5	6	7
8	5	9	7	6	1	4	2	3
4	2	6	8	5	3	7	9	1
7	1	3	9	2	4	8	5	6
9	6	1	5	3	7	2	8	4
2	8	7	4	1	9	6	3	5
3	4	5	2	8	6	1	7	9

What you need to do?

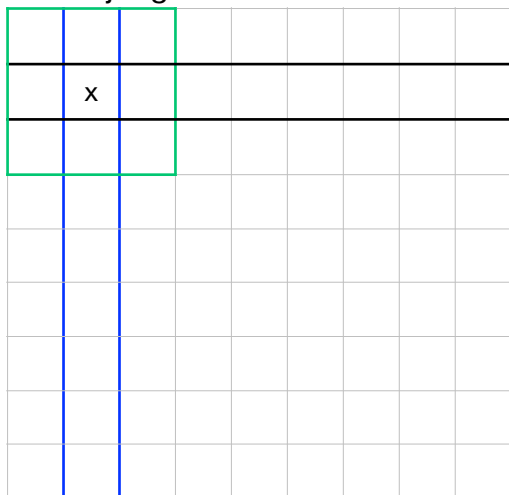
Each time you finish the following 4 judgements, there is **at least** one number can be filled in, and keep running those judgements until you find the solution. That is, you **DO NOT** need to use more complicated logic and guess the numbers in this assignment.

1. The judgement of the blocks in the column:



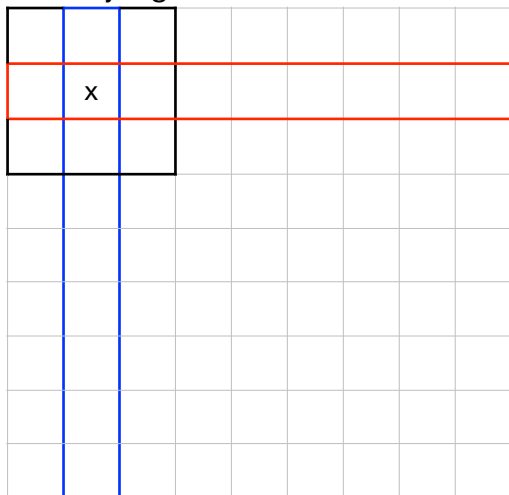
```
for each missing number  $n$  in the column do
  for each blank block in the column do
    if the corresponding row and subgrid
      doesn't exist  $n$  do
      mark the block
    end if
  end for
  if there is only one blank block may fill  $n$  do
    fill  $n$  in that blank block
  end if
end for
```

2. The judgement of the blocks in the row:



```
for each missing number  $n$  in the row do
  for each blank block in the row do
    if the corresponding column and subgrid
      doesn't exist  $n$  do
      mark the block
    end if
  end for
  if there is only one blank block may fill  $n$  do
    fill  $n$  in that blank block
  end if
end for
```

3. The judgement of the blocks in the subgrid:



```
for each missing number  $n$  in the subgrid do
  for each blank block in the subgrid do
    if the corresponding row and column
      doesn't exist  $n$  do
      mark the block
    end if
  end for
  if there is only one blank block may fill  $n$  do
    fill  $n$  in that blank block
  end if
end for
```

4. The judgement of the last number:

Check whether there is only one blank block that is not filled in each column, each row and each subgrid. If it is, fill the last missing number.

Example of more complicated logic:

The above judgements can only solve the left sudoku puzzle to the right sudoku puzzle.

				6	8		3	
1	9							
8		3	1			2		
4				5	1		6	
7				2				4
				7		8		
	1				5			7
		4						
	5			3		1		



	4	7		6	8	9	3	1
1	9			4		6	7	8
8	6	3	1	9	7	2	4	5
4				5	1	7	6	
7				2		3		4
			4	7		8		
3	1			8	5	4		7
	7	4		1		5	8	3
	5	8	7	3	4	1		6



	4	7		6	8	9	3	1
1	9			4		6	7	8
8	6	3	1	9	7	2	4	5
4				5	1	7	6	
7	8			2		3		4
			4	7		8		
3	1			8	5	4		7
	7	4		1		5	8	3
	5	8	7	3	4	1		6

Consider the 2nd column:

The missing numbers are 2,3 and 8. By the above judgements, 2 and 3 can be filled in 4th and 6th blocks and 8 can be filled in 4th and 5th blocks, so it is impossible to find out the proper number to fill in this column. However, 8 can be filled in 5th block. Since it is not possible to fill 2 and 3 into the fifth blank block. Because there are both 2 and 3 in the 5th row and 8 exists in the 6th row, the only place can fill 8 is 5th blank block.

Therefore, you don't need to **record** each blank block that what numbers can fill in and then go back to check if it is the proper number or not. You **ONLY** need to approach the 4 judgements to finish this assignment, and you don't need to follow the pseudo code if you have other logic to complete this assignment. You may think that how to use more complete logic to solve ALL sudoku puzzles.

Input and Output Format

The input format:

1. A space after each number and blank block, but there is no space after the last number of each row.
2. Press the enter after the last number of each row.

The output format:

1. A space after each number, including the last number in each row.
2. Change a new line at the end of each row.

Sample Input and **output** 1:

```
5 3      7
6      1 9 5
  9 8      6
8      6      3
4      8 3      1
7      2      6
  6      2 8
    4 1 9      5
      8      7 9
5 3 4 6 7 8 9 1 2
6 7 2 1 9 5 3 4 8
1 9 8 3 4 2 5 6 7
8 5 9 7 6 1 4 2 3
4 2 6 8 5 3 7 9 1
7 1 3 9 2 4 8 5 6
9 6 1 5 3 7 2 8 4
2 8 7 4 1 9 6 3 5
3 4 5 2 8 6 1 7 9
```

Sample Input and **output** 2:

```
  7      2 6 1
9 6      8
    5      6
  9 6      1
2 1 8      7 5
    4 2      6
5      6 7 8 4
      3 2
8      1 3
3 7 4 8 9 2 5 6 1
9 6 2 5 4 1 3 7 8
1 8 5 3 7 6 4 2 9
4 9 6 7 1 5 8 3 2
2 1 8 6 3 9 7 4 5
7 5 3 4 2 8 1 9 6
5 3 1 2 6 7 9 8 4
6 4 9 1 8 3 2 5 7
8 2 7 9 5 4 6 1 3
```

NTHU Online Judge information:

Problem ID: 11738

Problem title: 231001_12/31_Assignment2

Guidelines

1. Mark weightings: Correctness 50% + Readability 50%.
 - A. Correctness: Make sure you understand what the program should do in every case (including special cases).
 - B. Readability: Use comments to explain your logic.
2. You are welcome to discuss with each other, but **DO NOT COPY OTHER PEOPLE'S WORK**. Plagiarism is a serious offense. Not only will you get no points in this assignment, but you may also be reported to the university.
3. Program Style:
 - A. Your program should include a number of functions. Their functionality should be well-defined, easily understandable, and clearly documented as comments within the source code.
 - B. The efficiency of your program should be reasonable. However, don't spend too much time just to speed it up while making the code difficult to read.