Problem 2 (Marks: 20)

Handed out: 17 September
Due: 20 September 8:30 am

You might have heard the name of Sudoku - a game of logic involving numbers.
Consider the puzzle in Figure 1. The objective is to fill this 9x9 grid in such a way that each **column**, each **row**, and each of the nine **3x3 boxes** therein contain each of the numbers 1 through 9 exactly once. Calm Down Now!!! You don't have to solve a Sudoku.

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

Figure 1

Rather, your task is check if a given Sudoku configuration is valid or not in main function. Read the instructions provided below carefully.

First, open the main.c file that has been provided to you. You will find 2 functions there – i) a main function as usual and ii) a user defined function. The prototype of the user defined function looks like this:

```
int check(int x[], int size);
```

It will take a 1d array and its size as input and will return 1 if the input 1d array contains 1- 9 each digit exactly once or 0 if it does not. It has already been implemented for you.
In no way you should edit this function or use any other user defined function for solving this problem.

Now the main function. Your task is to actually implement this main function. First, read the 81 cells of the input Sudoku and then call the above user-defined function multiple times for checking if the input Sudoku is configured correctly or not.

The input Sudoku will require processing to pass it to the user defined function, e.g. you have to check each row of the Sudoku to see if it contains 1-9 exactly once.
For that, you might want to create a 1d array for 0^{th} row (1, 6, 8, 3, 4, 2, 7, 9, 5) and pass it to user-defined function check to see if it returns 1 or not.

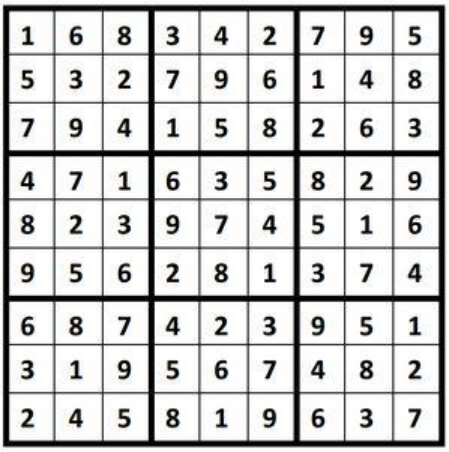
Input

Read 81 cells of Sudoku row-wise (First take the cells of 0^{th} row, then 1^{st} row and so on). Each cell must have values in the range of 1 – 9.

Output

If the configuration is valid print valid else print not valid

Sample Input & Output



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

Enter 81 cells:
1 6 8 3 4 2 7 9 5
5 3 2 7 9 6 1 4 8
7 9 4 1 5 8 2 6 3
4 7 1 6 3 5 8 2 9
8 2 3 9 7 4 5 1 6
9 5 6 2 8 1 3 7 4
6 8 7 4 2 3 9 5 1
3 1 9 5 6 7 4 8 2
2 4 5 8 1 9 6 3 7

Output: valid

Enter 81 cells:
1 6 8 3 4 2 7 9 5
5 3 2 7 9 6 1 4 8
7 9 4 1 5 8 2 6 3
4 7 1 6 3 5 8 2 9
8 2 3 9 7 4 5 1 6
9 5 6 2 8 1 3 7 4
6 8 7 4 2 3 9 5 1
3 2 9 5 6 7 4 8 2
2 4 5 8 1 9 6 3 7

Output: Not valid

Collaboration

In no circumstances you are allowed to share your code with others. If a case arises as such both the provider and receiver will get straight zero. However, you can share ideas on how to approach the problem.

Submission

Submit your file as **Your_10_digit_sudent_id.c**, e.g. if a student's id is 2013-3-60-23 his code should be saved as 2013-3-60-23.c