Project #3: Sudoku checker

Task description:

According to Wikipedia: *"Sudoku is a logic-based, combinatorial number-placement puzzle. In classic sudoku, the objective is to fill a 9×9 grid with digits so that each column, each row, and each of the nine 3×3 subgrids that compose the grid (also called 'boxes', 'blocks', or 'regions') contain all of the digits from 1 to 9. The puzzle setter provides a partially completed grid, which for a well-posed puzzle has a single solution."*

The example below illustrates the nine blocks from a completed sudoku puzzle:

```
1 3 2 | 5 7 9 | 4 6 8
4 9 8 | 2 6 1 | 3 7 5
7 5 6 | 3 8 4 | 2 1 9
------+-------+------
6 4 3 | 1 5 8 | 7 9 2
5 2 1 | 7 9 3 | 8 4 6
9 8 7 | 4 2 6 | 5 3 1
------+-------+------
2 1 4 | 9 3 5 | 6 8 7
3 6 5 | 8 1 7 | 9 2 4
8 7 9 | 6 4 2 | 1 5 3
```

Your task is to read a completed sudoku puzzle and check whether it is a valid solution or not. In a valid solution:

1. Each line contains all digits from 1 to 9.
2. Each column contains all digits from 1 to 9.
3. Each block contains all digits from 1 to 9.

Input specification:

The input of a test case contains nine lines, and each line contains nine integers in the range [1,9].

Output specification:

Your program must print a single line as output. Print **YES** if the input is a valid sudoku solution, and **NO** otherwise. Do not forget the new-line character in the end.

Example #1:

Input	Output

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

Example #2:

Input	Output

3 1 2 5 7 9 4 6 8
4 9 8 2 6 1 3 7 5
7 5 6 3 8 4 2 1 9
6 4 3 1 5 8 7 9 2
5 2 1 7 9 3 8 4 6
9 8 7 4 2 6 5 3 1
2 1 4 9 3 5 6 8 7
3 6 5 8 1 7 9 2 4
8 7 9 6 4 2 1 5 3
	NO