

SuDoku Puzzle

The name "Sudoku" is the Japanese abbreviation of a longer phrase, "suji wa dokushin ni kagiru", meaning "the digits must occur only once". Sudoku is a logic-based number placement puzzle. The objective is to fill a 9x9 grid so that each column, each row, and each of the nine 3x3 boxes contains the digits from 1 to 9. The puzzle setter provides a partially completed grid.

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

Unlike in magazines and newspapers, the digital representation of Sudoku a puzzle is a string of length 81, with all rows of the puzzle placed one after another. The representation uses ASCII symbols '1'-'9' for digits and '.' for an empty space. For example, the puzzle from figure above can be represented as:

7..25..98..6....1....61.3..9....1.....8.4.9..75.28.1.94..3.....4923.61.....4.

In this task you are to solve such puzzles automatically. The score will depend on the number of solved puzzles and on the speed of your solution. Some of the puzzles have multiple possible solutions, so be careful. A solution is correct if it satisfies the given puzzle. You can be sure that all given Sudokus are correct.

Input

t – the number of test cases; then t test cases follows. [$t \leq 500$]

Each test case describes one SuDoku puzzle and consists of an 81-character-long string.

Output

For the i -th test case output a line containing **Y** if you want to solve the test case or **N** if you wish to leave it out. If you chose to solve the test case, in the next line output a sequence of exactly 81 characters corresponding to the solution for the i -th Sudoku puzzle.

Score

The score for this task calculated using the formula: $score = 200 * total_solved / (200 + time)$, where $total_solved$ - number of correctly solved puzzles, $time$ - running time of your program. If the score

has the following form: $xxx.xxxaaa$, then aaa - is the number of correctly solved puzzles.

Example

Input:

3

..41..3.8.1....62...82..4.....3.28.9....7....7.16.8...562..17.3.3.....4.1....5...
1.....4....1.38.27.9.4...91.7.....5..86.4.5.9..3.....8..9....2.4.....7
7..25..98..6....1....61.3..9....1.....8.4.9..75.28.1.94..3.....4923.61.....4.

Output:

Y

294167358315489627678253491456312879983574216721698534562941783839726145147835962

Y

198563274654217389273984615915726843347198562862435791731642958589371426426859137

N

Score:

In this case $total_solved = 2$. If the program runs for 10 seconds, then the score of this solution will be equal to 1.905002