

Sudoku

Sudoku is a game of logic where numbers have to be placed in a table according to certain rules. In this exercise we use a table of 9×9 fields.

The table is divided into 9 3×3 -sized regions as shown in the figure below. Each region has to be filled with numbers 1, 2, 3, 4, 5, 6, 7, 8, 9 so that each row and each column of the 9×9 -sized table contains each of numbers 1...9 exactly once. The creator of the puzzle fills certain cells of the table in advance. The solver's task is to fill the remaining cells according to the rules above.

Each of the first 9 lines of the input text files contains 9 integers representing the initial state of the puzzle. The fields that are not filled contain number 0. The next few lines contain a few attempted steps by the solver to fill a field. A step is described by three numbers: the number the solver wishes to enter, then the row and column identifiers of the field where they wish to enter the number. In the lines of the input file the numbers are separated from each other by single spaces. The table is free from contradictions, so it describes a puzzle that has a solution. The number of steps attempted by the solver is at least 1, at most 10, some of them may be invalid.

Example:

```
0 0 0 0 0 0 0 0 6
0 0 7 0 0 0 2 0 0
3 8 9 5 0 0 0 0 0
6 0 0 0 0 0 7 0 0
1 2 8 4 9 0 0 3 0
0 0 0 0 0 0 0 0 9
0 5 1 0 0 0 0 4 0
0 0 0 0 0 3 0 9 1
0 0 0 0 8 0 0 0 0
9 2 4
1 2 1
5 9 9
7 2 2
```

0	0	0	0	0	0	0	0	6
0	0	7	0	0	0	2	0	0
3	8	9	5	0	0	0	0	0
6	0	0	0	0	0	7	0	0
1	2	8	4	9	0	0	3	0
0	0	0	0	0	0	0	0	9
0	5	1	0	0	0	0	4	0
0	0	0	0	0	3	0	9	1
0	0	0	0	8	0	0	0	0
9	2	4						
1	2	1						
5	9	9						
7	2	2						

The above example shows the contents of input file *difficult.txt*. The numbers in the 10th line mean that 9 should be entered in the 4th position of row 2. None of the given row and given column and the given region contains 9, so the step does not result in an error, it is valid. The step given in the 11th line is also valid. The step in the 13th line is an invalid step, because row 2 already contains number 7.

The numbers in circles in the right-hand figure show the identifiers of the 3×3 -sized regions.

Create a program that uses one of the input files (*easy.txt*, *medium.txt*, *difficult.txt*) to answer the following questions. Save the source code of the program as *sudoku*. (When writing the program, you do not have to check the correctness and the validity of the data given by the user, you can assume that the data available correspond to the description.)

When an exercise part requires displaying information on the screen, display the exercise number on the screen (for example **Exercise 4.**). If you request data from the user, display the nature of data requested. Displaying information without accents is acceptable.

1. Request the name of a file, the identifier of a row and the identifier of a column (numbers between 1 and 9). Solve the following exercises using these values.
2. Read the contents of the file whose name was given in the previous exercise and store the data of the table. If it is not possible, then use one of the available files as a source.

3. Display on the screen
 - a. the value stored in the position with the entered row and column identifiers. If the value in the given position is 0, then display text “The given position is not filled yet.”
 - b. the region the position with the entered row and column identifiers belongs to.
4. Determine what percent of the table is not filled yet. Display the result on the screen with an accuracy of one decimal figure.
5. Examine whether the steps given in the file are valid in the read table. Consider each step as if it was the only step made in the original table but do not execute it. Determine whether the execution of the steps results in a contradiction. Display the three values belonging to the step, then display one of the following statements in the next line. If several statements are true, it is enough to display one of them.
 - “The position is already filled.”
 - “The given row already contains the number.”
 - “The given column already contains the number.”
 - “The given region already contains the number.”
 - “The step is valid.”

Example for the format of textual output:

```
Exercise 1.  
Enter the name of the input file: easy.txt  
Enter a row identifier: 1  
Enter a column identifier: 1  
  
Exercise 3.  
The number in the given position is 5  
The position belongs to region 1.  
  
Exercise 4.  
The ratio of blank positions is 17.3%  
  
Exercise 5.  
The selected row: 2, column: 4, number: 9  
The position is already filled.  
  
The selected row: 3, column: 6, number: 7  
The step is valid.  
  
The selected row: 6, column: 6, number: 3  
The given region already contains the number.  
  
The selected row: 7, column: 9, number: 8  
The given column already contains the number.
```