

# Quick answer

[English](#)

[Vietnamese](#)

Joe is fond of computer games. Now, he must solve a puzzling situation. In front of his eyes lies a huge map with fortified towns. His enemy is a very powerful and tricky character who can connect and disconnect the towns by giving some commands. Two towns are connected if they have been directly connected or interconnected through some other connected towns at some moment in time. When a town is disconnected it gets isolated and clears its own connection history, not the connection history of the other towns. Each connection is bi-directional. Initially the towns are isolated. Joe is asked to answer quickly if two towns are connected, according to the history of the character's commands. Write a program which based on information input from a text file counts the number of yes answers and the number of no answers to questions of the kind: is town  $i$  connected with town  $j$ ?

## Input

The program reads data from a text file. Each data set in the file stands for a particular map and the associated character's commands, as follows:

- 1) The number of towns on the map  $N$  ( $N \leq 10000$ );
- 2) A list of commands of the form:
  - a)  $c$  town $i$  town $j$ , where town $i$  and town $j$  are integers from 1 to no\_of\_towns. The command means that town $i$  and town $j$  get connected.
  - b)  $d$  town $i$ , where town $i$  is an integer from 1 to no\_of\_towns. The command means that town $i$  gets disconnected.
  - c)  $q$  town $i$  town $j$ , where town $i$  and town $j$  are integers from 1 to no\_of\_towns. The command stands for the question: is town $i$  connected with town $j$ ?
  - d)  $e$ , that ends the list of commands

Each command is on a separate line. Commands (a), (b), (c) can appear in any order.

The towns' connectivity is updated after each command of type (a) or (b).

Each command of type (c) is processed according to the current configuration.

## Output

Assume there are  $N_1$  yes answered questions and  $N_2$  no answered questions.

The program prints these two numbers to the standard output on the same line, in the order:  $N_1$   $N_2$  as shown in the figure:

## Sample

For example, the input file illustrated in the figure below corresponds to only one data set which stands for a map with 4 fortified towns.

The character gives 9 commands.

Input :

4

c 1 2  
c 3 4  
q 1 3  
c 2 3  
q 1 4  
d 2  
q 4 1  
q 2 4  
e  
Ouput:  
2 2