

6563 Spectrum

Swamp County Consulting has just been awarded a contract from a mysterious government agency to build a database to investigate connections between what they call “targets.” Your team has been sub-contracted to implement a system that stores target information and processes the commands listed below.

A *target* is represented by a string of up to 32 printing characters with no embedded spaces. A *connection* is a bi-directional relationship between two targets.

The *hop count* between a given target (called “target1”) and other targets is determined by the following rules:

1. Targets directly connected to *target1* are 0 hops away.
2. Targets directly connected to the 0 hop targets, and not already counted as a 0 hop target or the original target, are 1 hop targets.
3. Similarly, targets directly connected to n hop targets, and not already counted in $0 \dots n$ hop targets are $n + 1$ hop targets.

There will be no more than 100,000 targets and no more than 500,000 connections.

Commands

The data base system has only three commands: **add**, **associated**, and **connections**. Targets and connections are never deleted because the Agency never forgets or makes mistakes. Commands start in the first column of a line. Commands and their parameters are separated by whitespace. No input line will exceed 80 columns. No leading or trailing whitespace is to appear on an output line.

add takes one or two parameters:

add *target1* //single parameter

Function: Adds the target to the database, with no connections.

Note: If target is already in the database, do nothing (not an error)

add *target1 target2* //two parameters

Function: Creates a bidirectional connection between the targets.

- If either target is not yet in the database, add it/them, and create the connection.
- If there is already a connection between the targets, do nothing (not an error)
- If *target1* and *target2* are the same, treat this as if the command were “**add target1**” (not an error) There can be at most one direct connection between targets.

connections *target1*

Function: Returns the number of hops to direct and indirect connections from a target.

- Print the hop count, a colon, a single space, and the number of targets with that hop count with no leading zeroes on a separate line for each hop count. Start with hop count 0 and end with the hop count for the last nonzero number of targets. Do not print trailing spaces.
- If the target has no connections, print a line containing only the string ‘no connections’.

- If the target is not in the database, print a line containing only the string ‘target does not exist’ (no period).

associated *target1 target2*

Function: Returns information about the existence of a connection between the two targets

- If there is a path between the targets, print ‘yes: *n*’ on a separate line, where *n* is the hop count of *target2* with respect to *target1*. There is one space after the colon and no leading zeroes and no trailing spaces.
- If there is no connection between the targets, print ‘no’ on a separate line.
- If either *target1* or *target2* is not in the database, print a line containing only the string ‘target does not exist’.

Input

To allow for multiple test cases on the input file, we add a command ‘reset’ that is not a database command and occurs between the database commands for the different cases on the input file. Be sure to reset all of your data structures when you read this command. Process until end of file; there is no end-of-data flag and no ‘reset’ command at the end of the file.

Output

Start each case with a line consisting of ‘Case’, one space, the case number, and a colon. End each case with a line of ten minus signs.

Sample Input

```
add a b
add a c
add b d
add e b
add c f
add c g
add f h
add h i
add j k
associated a i
associated i a
associated f k
associated a h
connections a
connections i
add k g
associated a j
connections a
add h a
connections a
associated a h
add m
add n n
connections n
```

```
add a n
connections n
```

Sample Output

```
Case 1:
yes: 3
yes: 3
no
yes: 2
0: 2
1: 4
2: 1
3: 1
0: 1
1: 1
2: 1
3: 2
4: 1
5: 2
yes: 3
0: 2
1: 4
2: 2
3: 2
0: 3
1: 5
2: 1
3: 1
yes: 0
no connections
0: 1
1: 3
2: 5
3: 1
4: 1
-----
```