

SWE2004: Principles in Programming(Spring 2014)

Lab #11

Due-date: May 29, 14:45 PM

Q&A: jminlee at csl.skku.edu

Description

Rio de Janeiro is a very beautiful city, but there are so many places to visit that sometimes you feel overwhelmed, Fortunately, your friend Bruno has promised to be your tour guide.

Unfortunately, Bruno is terrible driver. He has a lot of traffic fines to pay and is eager to avoid paying more. Therefore he wants to know where all the police cameras are located so he can drive more carefully when passing by them. These cameras are strategically distributed over the city, in locations that a driver must pass through in order to travel from one zone of the city to another. A location C will have a camera if and only if there are two city locations A and B such that all paths from A to B pass through a location C.

For instance, suppose that we have six locations (A, B, C, D, E, and F) with seven bidirectional routes B - C, A - B, C - A, D - C, D - E, E - F, and F - C. There must be a camera on C because to go from A to E you must pass through C. In this configuration, C is the only camera location.

Given a map of the city, help Bruno avoid further fines during your tour by writing a program to identify where all the cameras are.

Input

The input will consist of an arbitrary number of city maps, where each map begins with an integer N ($2 < N \leq 100$) denoting the total number of locations in the city. Then follow N different place names at one per line, where each place name will consist of least one and at most 30 lowercase letters. A non-negative integer R then follows, denoting the total routes of the city. The next R lines each describe a bidirectional route represented by the two places that the route connects.

Location names in route descriptions will always be valid, and there will be no route from one place to itself. You must read until N = 0, which should not be processed.

Output

For each city map you must print the following line:

City map #d: c camera(s) found

where d stands for the city map number (starting from 1) and c stands for the total number of cameras. Then should follow c lines with the location names of each camera in alphabetical order. Print a blank line between output sets.

Sample Input

6

sugarloaf

maracana

copacabana

ipanema

corcovado

lapa

7

ipanema copacabana

copacabana sugarloaf

ipanema sugarloaf

maracana lapa

sugarloaf maracana

corcovado sugarloaf

lapa corcovado

5

guanababay

downtown

botanicgarden

colombo

sambodromo

4

guanababay sambodromo

downtown sambodromo

sambodromo botanicgarden

colombo sambodromo

0

Sample Output

City map #1: 1 camera(s) found

sugarloaf

City map #2: 1 camera(s) found

sambodromo