Problem 5: Minions

Overview

Imagine an asymmetric social network (think Twitter) of “Minions” in which members may follow other members, but the relationship is not reciprocated. Periodically, members leave this network. When they do, all of their followers simultaneously quit the network. (This property is not propagated; if B follows A and C follows B but not A, then when A leaves, B will also leave, but C will remain.)

Problem

In this problem you will be given an initial social network and a number N. What will happen is that the most popular Minion in the network (or the one with the lowest ID in the case of ties), will leave the network as will all of that Minion’s followers. This will happen N times – each time the most popular remaining Minion will leave, taking all of its followers. You are to determine the size of the network that remains. (Note that if the network is empty, that is not an error; it will simply be the case that no one leaves and thus the network is unchanged.) You are to report the size of the network after the N “defections” have taken place.

Input

The input to this problem comes in two lines. The first line contains the name of the network. Network names consist of four to sixteen (inclusive) letters (either case), digits, or spaces. The second line consists of a single integer, N, representing the number of successive defections. N will be between 0 and 100 inclusive.

Output

The output has a single line of text, formatted as in the examples, stating the number of Minions remaining after N defections. (Note that if there is only one Minion, the grammar is slightly off; see Example 2.)
About the class
You must use the SocialNetwork and Minion classes in your solution to this problem. Otherwise, you will not be able to access the network or any of the Minions contained therein. All of the necessary methods are provided for you (although you may not need to use everything that is provided). Be sure to pay close attention to documentation of the class, including the pre- and post-conditions for all of the methods. Note: any network whose name begins with “factor” will be structured so that the following relation is based upon factoring, i.e. Minion #x will follow Minion #y if and only if x is a factor of y. And, for the egoists in the group, a Minion can follow itself!

Example 1 (Note: neither of the examples has leading or trailing spaces in the network name)

Input

    factor network
    5

Output

    There are 12 Minions in the network.

Example 2

Input

    abracadabra
    11

Output

    There are 1 Minions in the network.