UCF "Practice" Local Contest — Aug 27, 2016

The Eternal Quest for Caffeine

filename: `soda`
(*Difficulty Level:* `Hard`)

Like most engineering students, Jason relies on caffeine, soda in particular, to get him through long nights in the lab. He's not too picky about the brand; as long as it's fizzy, loaded with caffeine, and not the "diet" version, he's happy. When the new Harris Engineering Center opened on the UCF Campus, he was pleased to see a soda machine on every floor. Not just any soda machine, either. These machines are the fancy kind that display all of the soda stock arranged in rows stacked on top of each other (like most snack machines). When a soda is purchased, a conveyor belt rises to the correct row, where the soda is surgically picked up by a robotic clasp that can travel left and right on the conveyor. The conveyor then descends to the vending slot, where the clasp gently deposits the soda. Finally, the slot unlocks and tilts forward, allowing the buyer to retrieve his or her soda. Engineering perfection! And as a bonus, the soda isn't subjected to all the usual mechanical clatter that causes it to fizz over when it's opened.

Unfortunately, these elaborate machines seem to have a propensity for component failure. On one soda mission from his lab, Jason discovered that the vending slot was broken on the machine on his floor, which prevented it from working altogether. He went to the next floor down and saw that that machine's vending slot was fine, but the conveyor was broken. He went down to the ground floor and saw that that machine was in perfect order, but only had caffeine free diet soda! At this point, Jason devised a plan. It's a simple matter for him to open up the working machine and harvest the parts he needs for the machine upstairs, then to hike back upstairs and repair the machine that houses the soda he needs. Sure, he *could* just take the soda he wants while the machine is open, but what fun would that be?

The one issue with this plan is that while Jason does enjoy the engineering challenge, he hates the walking between various broken machines each time he goes to get a coke, so he's asked you, a computer science student and fellow resident of the Harris Engineering Center to help. He can devise a way to monitor each machine in the building and ascertain what parts are working. He needs you to write a program that will allow him to get all the parts he needs from the various machines in the building, traveling up and down as few flights of stairs as possible (he doesn't trust the elevators because he's never been allowed to see how they work). Assume he can carry an unlimited number of parts. He also wants this algorithm to work for various kinds of coke machines and various buildings, in case the vendors decide to change out their machines one day, or the administration decides to relocate the EECS department again (you still can assume that there will always be exactly one coke machine on each floor).

The Problem:

Given the number of floors in the building, the number of parts required for a working machine, a description of the working parts in each machine in the building, and whether or not each machine has the desired kind of soda, determine the smallest number of floor changes required to

assemble a working machine that is already stocked with the desired soda. Jason will always start from his lab and return there after getting his soda.

The Input:

There will be multiple soda machine arrangements to process. Input will begin with three integers, *N*, *F*, and *P* ($1 \leq N,F,P \leq 10$), each separated by a single space with no leading or trailing spaces. *N* describes the number of floors in the building, *F* indicates which floor Jason's lab is on, and *P* indicates the number of different parts in each of the building's soda machines.

On the next *N* lines will be a set of integers followed by a single letter. Each line describes the soda machine on one floor (starting with the ground floor, and proceeding upward in order). The characters on a line are separated by a single space, with no leading or trailing spaces. The first integers on each line will be *S* ($0 \leq S \leq P$), indicating the number of working parts in the machine. *S* integers will follow, each indicating a working part in the machine (each of these integers will be unique and will be between 1 and *P*). Finally, there will be a single character "`Y`" or "`N`", where "`Y`" indicates that the machine has a kind of soda that Jason likes, and "`N`" indicates that it does not.

End of input will be indicated by a value of 0 for *N*, *F*, and *P*. This case should not be processed.

The Output:

For each soda machine arrangement, print the case number (starting with 1) and a single integer indicating the minimum number of times Jason will have to travel up or down a staircase to collect the parts he needs to repair a soda machine, get a soda that he wants, and return to his lab. If there is no way for Jason to get a soda, print "`Impossible`" instead of the integer. Leave a blank line after the output for each test case.

Sample Input:

```
4 2 5
5 1 2 3 4 5 N
4 1 2 3 5 Y
4 1 2 4 5 Y
5 1 2 3 4 5 Y
4 2 6
1 1 Y
2 2 3 Y
3 1 4 5 Y
0 Y
0 0 0
```

Sample Output:

```
Test case #1: 2

Test case #2: Impossible
```