

UCF Local Programming Contest — August 30, 2014

Chain Email

filename: email

(*Difficulty Level:* Medium)

A chain email is an email that people receive and then forward to all of their friends. This sort of email is very common amongst elderly people, who have notably bad memories. Elderly people's memories are so bad that if they ever receive a chain email they will forward it to all of their contacts. This can become very problematic when elderly people continually send the same email to each other. For instance, if two people have each other in their contacts and if either of them receive a chain email they will continually send the email to each other back and forth forever. Email companies are worried that this will result in a massive amount of storage loss on their servers and have asked you to determine if a specific person were to start a chain email, who would receive that email forever.

The Problem:

Given each elderly person's contacts and which elderly person will be starting a chain email, determine who will be indefinitely receiving emails.

The Input:

The first line of the input is a positive integer, n , indicating the number of scenarios that your program will have to analyze. Following this will be the description of each scenario. The first line of each scenario will have two single-space-separated integers, p ($1 \leq p \leq 50$), indicating the number of people who use the email service and, s ($1 \leq s \leq p$), indicating the source of the chain email, where each person is labeled from 1 to p . Following this will be a single line with the names of all of the people, from person 1 to person p , who use the email service, each separated by exactly one space. All names will contain alphabetic characters only and be between 1 and 19 characters (inclusive) in length. Following this will be p lines. The i^{th} line will describe the contact list of the i^{th} person. This description will consist of an integer, m ($0 \leq m < p$), indicating the number of contacts this person has, followed by the 1-based index of each of the contacts, each separated by exactly one space. It's guaranteed that no one will contain themselves as a contact.

The Output:

The first line of the output for each scenario should be "Chain Email # d :", where d is the scenario number, starting with 1. Following this should be a line containing the names of all of the people who will infinitely receive chain emails, assuming that everyone continually forwards the email to all of their contacts. Each name should be followed by a space. List these contacts in the order that they appear in the input. If no one will infinitely receive chain emails, then print "Safe chain email!" instead.

Leave a blank line after the output for each data set. Follow the format illustrated in Sample Output.

Sample Input:

```
3
3 1
James Sarah John
2 2 3
2 1 3
2 1 2
3 1
James Sarah John
2 2 3
0
0
6 3
Ali Matt Glenn Sumon Arup Chris
2 3 5
0
1 4
1 1
1 2
2 5 4
```

Sample Output:

```
Chain Email #1:
James Sarah John
```

```
Chain Email #2:
Safe chain email!
```

```
Chain Email #3:
Ali Matt Glenn Sumon Arup
```