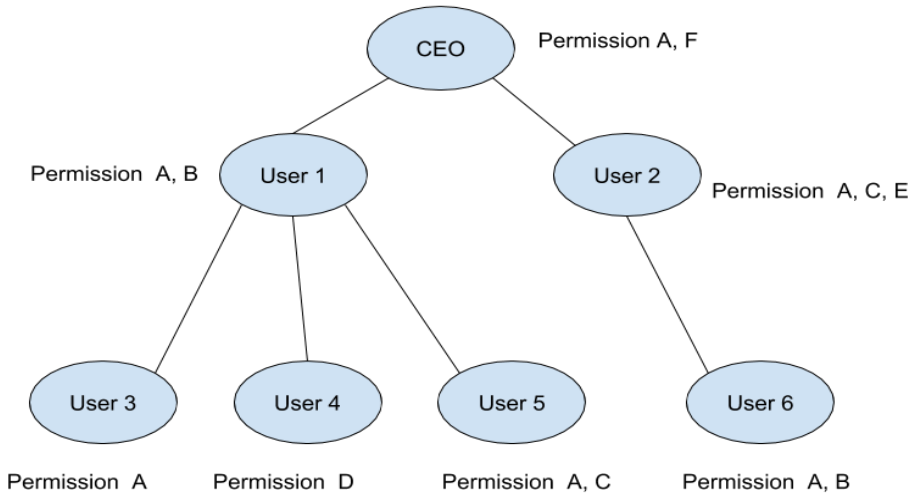
## Users & Permissions

Account management is one of the most fundamental features in most of the software. But a different application has a different requirement and its implementation can differ. Our application is for B to B and the client is a company. Let's think about a company as below.

In this company, all the users but CEO have a single manager to report and each manager can have staff who report to them and CEO has at least one staff to report to CEO. Each manager gets all permissions of staff under them in order to manage their work. It's guaranteed that all the users have at least 1 permission.

Let's say we have CEO and User 1-6 and permissions as below. In this case, User 1 manages User 3, User 4 and User 5 so User 1 will get permission A, B, C, D. User 2 will get permission A, B, C, E. CEO will have permission A, B, C, D, E, F. We would like to know which users have which permissions.




### Question 1

Please write the code to satisfy the requirements below. You can make assumption on the input as follows. The number of users n will be within:

0 < n < 100000

Each user can have up to k permissions:

0 < k < 100.

**Input / Output**

The input format is defined as below the number of users n is in the first line. And the next n lines are for each users' permission as below. For

the next (n - 1) lines, the manager for each user is placed. If the manager is CEO, string CEO is given but otherwise, the input should be number. The CEO doesn't have a manager to report.

### Input Format

```
<NUMBER OF USERS>
<CEO's PERMISSIONS>
<User1's PERMISSIONS>
...
<UserN's PERMISSIONS>
<User1's Manager>
...
<UserN's Manager>
```

You should output all the permissions on each line. The first line should be CEO. If a user has multiple permissions, the output should be concatenated with ','.

### Output Format

```
<CEO's ACCESSIBLE PERMISSIONS>
<User1's ACCESSIBLE PERMISSIONS>
...
<UserN's ACCESSIBLE PERMISSIONS>
```

***Example 1***

This is an example explained above.

### Input

```
6
A F
A B
A C E
A
D
A C
A B
CEO
CEO
1
1
1
2
```

**Expected Output**

```
A,B,C,D,E,F
A,B,C,D
A,B,C,E
A
D
A,C
A,B
```

*For Input method, we prefer system terminal, but any input method is still ok!

### Question 2

Write unit tests to verify you code behaviors. The test cases will also be checked.

### Question 3

Answer and explain the time & space complexity for your solution.
*Only code should be ok. Launching guide is preferable though.

### Question 4

Extend the program to support dynamic add/remove permissions and query individual user's permission. Additional inputs as follow:

```
Add 2 5
QUERY 2
REMOVE 2 2
QUERY 2
```

Explanation

These operations read as:

- ADD to user 2 permission 5. No output is expected.
- What is the permission of user 2 now? A list of permissions is expected. Output empty line if the user is having no permissions.
- REMOVE from user 2 permission 2. No output is expected.
- What is the permission of user 2 now? A list of permissions is expected. Output empty line if the user is having no permissions.

These input of these operations is right on the next line of the input of the original program. Input end at end of file (if using file) or when input wrong operation (anything that not start with ADD or REMOVE or QUERY, case sensitive).

Expected result :

```
1, 2, 3, 4, 5
1, 3, 4, 5
```