✏ Practice Arena

Practice problems aimed to improve your coding skills.

📂 PRACTICE-02_SCAN-PRINT

📂 PRACTICE-03_TYPES

📂 LAB-PRAC-02_SCAN-PRINT

📂 LAB-PRAC-01

📂 PRACTICE-04_COND

📂 BONUS-PRAC-02

📂 LAB-PRAC-03_TYPES

📂 PRACTICE-05_COND-LOOPS

📂 LAB-PRAC-04_COND

📂 LAB-PRAC-05_CONDLOOPS

📂 PRACTICE-07_LOOPS-ARR

📂 LAB-PRAC-06_LOOPS

📂 LAB-PRAC-07_LOOPS-ARR

📂 LABEXAM-PRAC-01_MIDSEM

📂 PRACTICE-09_PTR-MAT

📂 LAB-PRAC-08_ARR-STR

📂 PRACTICE-10_MAT-FUN

📂 LAB-PRAC-09_PTR-MAT

📂 LAB-PRAC-10_MAT-FUN

📂 PRACTICE-11_FUN-PTR

📂 LAB-PRAC-11_FUN-PTR

📂 LAB-PRAC-12_FUN-STRUC

     ❓ Point Pairing Party

     ❓ Verify the family tree of Mr C

     ❓ Simple Sodoku

     ❓ The Family Tree of Mr C Part Three

     ❓ The Post offices of KRville

     ❓ Matrix Mandala

     ❓ Mango Mania

     ❓ Recover the Rectangle

     ❓ Crazy for Candy

     ❓ A Brutal Cipher Called Brutus

     ❓ Triangle Tangle

     ❓ Basic Balanced Bracketing

📂 LABEXAM-PRAC-02_ENDSEM

📂 LAB-PRAC-13_STRUC-NUM

📂 LAB-PRAC-14_SORT-MISC

The Family Tree of Mr C Part Three

LAB-PRAC-12_FUN-STRUC

The Family Tree of Mr C Part Three [20 marks]

Problem Statement

In the first line of the input, we will give you n, a strictly positive number telling you how many members are there in Mr C's family. The family members will have names 1, 2, 3, 4, ..., n. In the next n-1 lines, we will tell you who is whose parent in this family by giving you pairs of numbers separated by a space i.e. in the format P C which tells us that P is the parent of C.

In Mr C's family there is one *head of the family* who has no parent. Everyone else has exactly one parent (it is a weird family where there is only one parent not two). However, people may have multiple children. The head of the family is said to belong to the *first generation* i.e. generation 1. There can be only one head of the family. All children of the head are said to belong to the *second generation* i.e. generation 2. All their children are said to belong to generation 3 and so on.

After giving you information about who is whose parent, we will give you a strictly positive number Q which tells you how many questions we are going to ask you. In the next Q lines we will give you names of Q family members (i.e. numbers between 1 and n, 1 and n included). In your output, you have to print to which generation do these family members belong.

Caution

1. Any number between 1 and n (both included) could be the head. The only distinguishing factor of the head is that the person has no parent
2. The parent child relationships will not be given to you in any particular order.
3. Your output should have Q lines too.
4. Be careful that we wont tell you who is the head's parent since the head has no parent. This is why we will only give you n-1 parent-child relationships.
5. Hint: if we do not give you the parent of a particular family member, that member has to be the head.
6. People may have multiple children but everyone has one and only one parent, except the head of the family who has no parent.
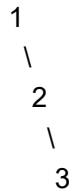7. Be careful about extra/missing lines and extra/missing spaces in your output.

EXAMPLE:
INPUT
3
1 2
2 3
2
1
2

OUTPUT:
1
2

Explanation: the family tree looks like

```
1
 \
  2
   \
    3
```

Since we did not give the parent of 1 in the input, 1 is the head. 1 is the parent of 2 and 2 is the parent of 3. 1 belongs to the first generation and 2 belongs to the second generation.

Grading Scheme:
Total marks: **[20 Points]**

There will be partial grading in this question. There are several lines in your output. Printing each line correctly, in the correct order, carries equal weightage. Each visible test case is worth 2 points and each hidden test case is worth 4 points. There are 2 visible and 4 hidden test cases.

Please remember, however, that when you press Submit/Evaluate, you will get a green bar only if all parts of your answer are correct. Thus, if your answer is only partly correct, Prutor will say that you have not passed that test case completely, but when we do autograding afterwards, you will get partial marks.

🍴 Start Solving! (/editor/practice/6233)