

# Яндекс. Тренировки по алгоритмам 2.0, занятие 8 (В)

## В. Родословная: предки и потомки

Ограничение времени	2 секунды
Ограничение памяти	64Mb
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

В генеалогическом древе у каждого человека, кроме родоначальника, есть ровно один родитель. Каждому элементу дерева сопоставляется целое неотрицательное число, называемое высотой. У родоначальника высота равна 0, у любого другого элемента высота на 1 больше, чем у его родителя.

Даны два элемента в древе. Определите, является ли один из них потомком другого.

### Формат ввода

Программа получает на вход число элементов в генеалогическом древе N. Далее следует N−1 строка, задающие родителя для каждого элемента древа, кроме родоначальника. Каждая строка имеет вид имя\_потомка имя\_родителя. Далее до конца файла идут строки, содержащие имена двух элементов древа.

### Формат вывода

Для каждого такого запроса выведите одно из трех чисел: 1, если первый элемент является предком второго, 2, если второй является предком первого или 0, если ни один из них не является предком другого.

### Пример

Ввод

Вывод

9  
Alexei Peter\_I  
Anna Peter\_I  
Elizabeth Peter\_I  
Peter\_II Alexei  
Peter\_III Anna  
Paul\_I Peter\_III  
Alexander\_I Paul\_I  
Nicholaus\_I Paul\_I  
Anna Nicholaus\_I  
Peter\_II Peter\_I  
Alexei Paul\_I

1 2 0

```

1 import sys
2
3 sys.setrecursionlimit(100000)
4
5
6 def search_parent(child, target, tree):
7     is_parent = False
8     while not is_parent and child in tree.keys():
9         if tree[child] == target:
10             is_parent = True
11         else:
12             child = tree[child]
13
14     return is_parent
15
16
17
18 # считываем данные
19 with open('input.txt', 'r') as f:
20     N = int(f.readline().strip()) # число элементов в генеалогическом древе
21
22     tree = dict() # генеалогическое древо ребенок:[родитель]
23     answer = []
24     # добавляем запись в генеалогическое древо, кроме родоначальника
25     for _ in range(N - 1):
26         child, parent = f.readline().split()
27         tree[child] = parent
28
29     for line in f.readlines():
30         human1, human2 = line.strip().split()
31         parent_1_2 = search_parent(human2, human1, tree)
32         parent_2_1 = search_parent(human1, human2, tree)
33         if parent_1_2:
34             answer.append(1)
35         elif parent_2_1:
36             answer.append(2)
37         else:
38             answer.append(0)

```

Отправить

Предыдущая

Следующая