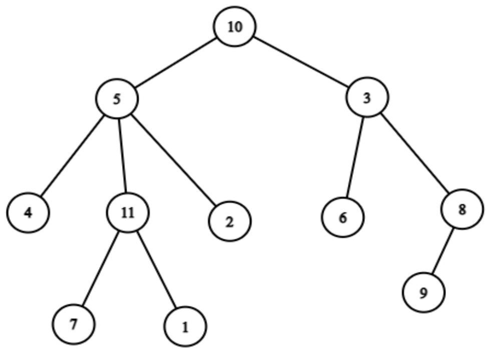
The following tree represent a tree of states each state with a unique ID where a state could be one of the following

a) A root state (In this Example its node number 10) which is the root of the tree (first node)
b) A parent state (In this Example 5,3,11,8) which is the node that have **at least one** children node (**Note that it's not a binary Tree, so parent could have many children)**
c) A leaf state (In this Example 4,7,1,2,6,9) which doesn't have any children



Considering the following structure (Only Example)

```
typedef struct node_s
{
    struct node_s * parent;
    uint8_t flag;
}node_t;
```

Represent a node in the above tree. You will be given a tree expressed as the following inputs:

The first line contains Integer number **(N)** the number of nodes in the tree where $3 \leq N \leq 10^3$

The N-1 next lines each line contain 3 integer numbers **S, P, F** $\mathbf{1 \leq S \leq 10^3}$, $\mathbf{1 \leq P \leq 10^3}$

$\mathbf{0 \leq F \leq 1}$ where **S** is the node ID, **P** is the parent ID, **F** the type of node

0: Leaf Node

1: Parent

(!!Note every node have a parent except the root node its parent ID is same as its own ID)

Next input line Contain Integer (K) the number of test cases $1 \leq K \leq 10^6$

Each following test case input line contains the **current active** leaf state node ID and the target State ID

Your task is to determine the path to make transition from the current active state to the target state

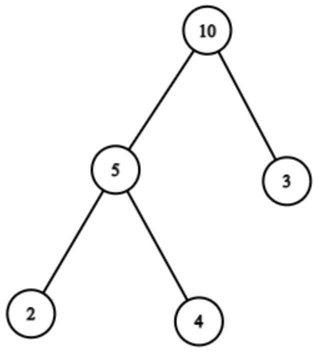
(!!Note that the active state can be only a leaf state, so it's guarantee that the target state always a leaf)

The Outputs is one line for each test case represent the path for the transition

Example: -

Inputs:

```
5
10  10   1
5    10   1
3    10   0
2     5   0
4     5   0
3
4     2
4     3
3     2
```



Output:

```
4    5   2
4    5    10   3
3    10   5    2
```