Celebrity

In this question, we define "Celebrity" in a party as a person who is known by everyone, but doesn't know anyone (Not count yourself of course). Input a data to represent who knows who such as `Ploy Pat`, meaning `Ploy` knows `Pat` (But doesn't mean `Pat` knows `Ploy`). For example, if input is..

```
Ploy Pat
Ploy Boy
Eak Pat
Boy Pat
Poom Pat
Boy Eak
```

It means that everyone (except `Pat`) knows `Pat` and `Pat` doesn't know anyone, meaning Pat is a celebrity.

A program below use **dict** name **R** with **key** as name and **value** contain a **set** of names that **key** knows. From example above, **dict** will be

```
R = {  'Ploy':{'Boy', 'Pat'},
       'Boy' :{'Pat', 'Eak'},
       'Pat' :set(),
       'Poom':{'Pat'},
       'Eak' :{'Pat'}              }
```

Write functions **knows, is_celeb** and **find_celeb** as shown in the comment below.

```python
def knows(R,x,y):
        # return True if x knows y

def is_celeb(R,x):
        # return True if a is celeb, otherwise return False
        # return False if x knows someone who is not him/herself
        # return False if there exists someone in R who don't know x
        # otherwise return True

def find_celeb(R):
        # for each person x in the party
        # if x is celeb --> return x
        # if no celeb in the party --> return None

def read_relations() :
        # build a dictionary R from inputs
        # whose structure is shown in the example

        R = dict()
        while True:
                d = input().split()
                if len(d) == 1 : break

                ???

        return R

def main():
        R = read_relations()
        c = find_celeb(R)
        if c == None :
                print('Not Found')
        else:
                print(c)

exec(input().strip()) # do not remove this line
```

Input

Command in Python language to test a function (test every functions I this question)

Output

Return output from a function call in input

Example

Input (from keyboard)	Output (on screen)

`main()`
`Ploy Pat`
`Ploy Boy`
`Eak Pat`
`Boy Pat`
`Poom Pat`
`Boy Eak`
`q`
	`Pat`

`main()`
`Ploy Pat`
`Ploy Boy`
`Eak Pat`
`Boy Pat`
`Poom Pat`
`Boy Eak`
`Noo-sa Tim`
`q`
	`Not Found`

`R ={'A':{'B'},'B':set()};print(knows(R,'A','B'),knows(R,'B','A'))`	`True False`