'''

This function takes two arguments,

data1 and data2, which contain

key-value pairs. All key-value

pairs within data1 are unique.

Similarly, all key-value pairs

within data2 are unique. However,

there may be key-value pairs (k, v1)

in data1 and (k, v2) in data2 with a

common key k. In this case, v1 and

v2 may be the same, or v1 and v2 may

be different.

This function should modify only

data1 and return a (possibly empty)

dictionary as follows:

For every key-value pair (k, v2) in

data2, if no key-value pair with key

k exists in data1, then the pair

(k, v2) should be added to data1.

Otherwise, there is a unique pair

(k, v1) already in data1. If v1 and

v2 are different, the pair (k, v1)

should be removed from data1 and the

key-value pair (k, [v1, v2]) should

be added to the (initially empty)

dictionary to be returned.

In this implementation, data1 is a

dictionary and data2 is a list where

each key-value pair in data2 is also

a list [key, value] of length 2.

'''

def uniqueUpdate(data1, data2):

    # Initially empty dictionary

    dupKeys = {}

    # Examine every (k, v2) pair in data2

    for k, v2 in data2.items():

        # Search for a key-value pair

        # with key = k in data1

        # (no such pair found yet)

        kFound = False

        for [k1, v1] in data1:

            if k1 == k:

                # Found pair with key = k

                kFound = True

                if v1 != v2:

                 # Remove (k, v1) from data1

                 data1.remove([k,v1])

                 # Add (k, [v1, v2])

                 # to dictionary

                 dupKeys[k] = [v1,v2]

        # After the loop, check if

        # k was not found

        if not kFound:

            # Add (k, v2) to data1

            data1.append([k, v2])

    # After processing all (k, v2) in

    # data2, return the dictionary

    return dupKeys

'''

Visualize this function on an example:

<https://tinyurl.com/dsaprac20>

'''

## DO NOT MODIFY BELOW THIS LINE! ##

'''

This part of the code reads input in

the following format:

Line 1: A positive integer n1

representing the number of key value

pairs in data1

Lines 2 to n1+1: Two integers k v

per line representing the key and

value (these n1 key value pairs are

added to data1)

Line n1+2: A positive integer n2

representing the number of key value

pairs in data2

Lines n1+3 to n1+n2+2: Two integers

k and v per line representing the

key and value (these n2 key value

pairs are added to data2)

This also prints the output in the

following format after calling the

uniqueUpdate function:

data1

data2 (should remain the same)

dup (the dictionary returned)

'''

import sys

if \_\_name\_\_ == '\_\_main\_\_':

    data1 = {}

    n1 = int(input())

    for \_ in range(n1):

        k, v = map(int, input().split())

        if k in data1:

            sys.exit("Illegal: data1")

        data1[k] = v

    data2 = []

    n2 = int(input())

    for \_ in range(n2):

        k, v = map(int, input().split())

        for [k2, v2] in data2:

            if k2 == k:

                sys.exit("Illegal: data2")

        data2.append([k, v])

    dup = uniqueUpdate(data1, data2)

    print(data1)

    print(data2)

    print(dup)