Input Format

The first line contains an integer, n, denoting the number of entries in the phone book.

Each of the n subsequent lines describes an entry in the form of 2 space-separated values on a single line. The first value is a friend's name, and the second value is an 8-digit phone number.

After the n lines of phone book entries, there are an unknown number of lines of queries. Each line (query) contains a $name$ to look up, and you must continue reading lines until there is no more input.

Note: Names consist of lowercase English alphabetic letters and are first names only.

Constraints

- $1 \leq n \leq 10^5$
- $1 \leq queries \leq 10^5$

Output Format

On a new line for each query, print `Not found` if the name has no corresponding entry in the phone book; otherwise, print the full $name$ and $phoneNumber$ in the format `name=phoneNumber`.

Sample Input

```
3
sam 99912222
tom 11122222
harry 12299933
sam
edward
harry
```

Sample Output

```
sam=99912222
Not found
harry=12299933
```

Explanation

We add the following $n = 3$ (Key,Value) pairs to our map so it looks like this:

$phoneBook = \{(sam, 99912222), (tom, 11122222), (harr$

We then process each query and print `key=value` if the queried key is found in the map; otherwise, we print `Not found`.

Query 0: `sam`

Sam is one of the keys in our dictionary, so we print `sam=99912222`.

Query 1: `edward`

Edward is not one of the keys in our dictionary, so we print `Not found`.

Query 2: `harry`

Harry is one of the keys in our dictionary, so we print `harry=12299933`.

Change Theme | Java 8

```
class Solution{
    public static void main(String []argh){

        Map<String, Object> phoneBook = new HashMap<>();

        Scanner in = new Scanner(System.in);
        int n = in.nextInt();
        for(int i = 0; i < n; i++){
            String name = in.next();
            int phone = in.nextInt();
            phoneBook.put(name, phone);
        }
        while(in.hasNext()){
            String s = in.next();
            String phone;
            try {
                 phone = phoneBook.get(s).toString();
            }catch (NullPointerException e){
                phone = null;
            }
            if (phone != null){
                System.out.println(s.concat("=").concat(
            }else System.out.println("Not found");
        }
        in.close();
    }
}
```

Line: 19 Col: 13

Upload Code as File
Test against custom input

Run Code | Submit Code

You have earned 30.00 points!

You are now 6 challenges away from the 3rd star for your 30 days of code badge.

25% 9/15

30 Days of Code

Congratulations

You solved this challenge. Would you like to challenge your friends?

The next challenge in this tutorial will unlock in 20:24:13

Go to Dashboard | Try a Random Challenge

Test case 0
Test case 1
Test case 2
Test case 3
Test case 4

Compiler Message

Success

Input (stdin) Download

```
3
sam 99912222
tom 11122222
harry 12299933
```