PRACTICE   COMPETE   JOBS   LEADERBOARD   🔍 Search   💬 🔔   👤 sashu1231 ⌄

Practice > Tutorials > 30 Days of Code > Day 8: Dictionaries and Maps   6 more challenges to get your next star!   **30** Days of Code ***   **14%** 16/22

Day 8: Dictionaries and Maps ☆

👤 **by** Shafaet

| Problem | Submissions | Leaderboard | Discussions | Editorial | Tutorial |

Objective

Today, we're learning about Key-Value pair mappings using a *Map* or *Dictionary* data structure. Check out the Tutorial tab for learning materials and an instructional video!

Task

Given n names and phone numbers, assemble a phone book that maps friends' names to their respective phone numbers. You will then be given an unknown number of names to query your phone book for. For each *name* queried, print the associated entry from your phone book on a new line in the form `name=phoneNumber`; if an entry for *name* is not found, print `Not found` instead.

Note: Your phone book should be a Dictionary/Map/HashMap data structure.

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:

$$phoneBook = \{(sam, 99912222), (tom, 11122222), (harry, 12299933)\}$$

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`.
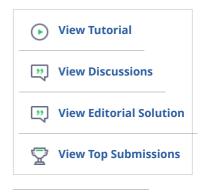
Query 1: `edward`
Edward is not one of the keys in our dictionary, so we print `Not found`.

Query 2: `harry`
Harry is one of the keys in our dictionary, so we print `harry=12299933`.

| Easy | Submitted 110085 times Max Score **30** |

Need Help?

▶ **View Tutorial**

💬 **View Discussions**

💬 **View Editorial Solution**

🏆 **View Top Submissions**

Rate This Challenge:

☆ ☆ ☆ ☆ ☆

Download problem statement

Download sample test cases

Suggest Edits

f   🐦   in

Current Buffer (saved locally, editable) ⑂ 🕘   Java 8 ⌄   ⤢   ⚙

```java
1  //Complete this code or write your own from scratch
2  import java.util.*;
3  import java.io.*;
4
5  class Solution {
6      public static void main(String[] argh) {
7          Scanner in = new Scanner(System.in);
8          int n = in.nextInt();
9          Map<String, Integer> fonebook = new HashMap<>();
10         for (int i = 0; i < n; i++) {
11             String name = in.next();
12             int phone = in.nextInt();
13             // Write code here
14             fonebook.put(name, phone);
15         }
16         while (in.hasNext()) {
17             String s = in.next();
18             // Write code here
19             if (fonebook.get(s) != null) {
```