



# 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 (in the form **name=phoneNumber**) or **Not found** if there is no entry for *name*.

**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 letters and are *first names* only.

## Constraints

- $1 \leq N \leq 10^5$
- $1 \leq \text{queries} \leq 10^5$

## 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

$N = 3$

We add the  $N$  subsequent (*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, or **Not found** otherwise.

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**.



Submissions: 2666

Max Score: 30

Difficulty: Easy

[More](#)

Current Buffer (saved locally, editable)  

Java 7  

```
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         for(int i = 0; i < n; i++){
10             String name = in.next();
11             int phone = in.nextInt();
12             // Write code here
13         }
14         while(in.hasNext()){
15             String s = in.next();
16             // Write code here
17         }
18         in.close();
19     }
20 }
21
```

Line: 1 Col: 1

 [Upload Code as File](#)

☐ Test against custom input

Run Code

Submit Code

Copyright © 2016 HackerRank. All Rights Reserved

Join us on IRC at [#hackerrank](#) on freenode for hugs or bugs.

[Contest Calendar](#) | [Blog](#) | [Scoring](#) | [Environment](#) | [FAQ](#) | [About Us](#) | [Support](#) | [Careers](#) | [Terms Of Service](#) | [Privacy Policy](#) | [Request a Feature](#)