Day 8: Dictionaries and Maps

by Shafaet

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

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: 48095
Max Score: 30
Difficulty: Easy

Rate This Challenge:

More

Current Buffer (saved locally, editable) C#

```
using System;
using System.Collections.Generic;
using System.IO;
class Solution {
    static void Main(String[] args) {
        /* Enter your code here. Read input from STDIN. Print output to STDOUT. Your class should be named Solution */

        int n = int.Parse(Console.ReadLine());

           Dictionary<string, string> diccio = new Dictionary<string, string>();
           for (int i = 0; i < n; i++)
           {
               string[] input = Console.ReadLine().Split(' ');
               diccio[input[0]] = input[1];
           }

           while (true)
           {
               string input = Console.ReadLine();
               if (input == null)
               {
                   break;
               }
               if (diccio.ContainsKey(input))
               {
                   //sam=99912222
                   Console.WriteLine(input + "=" + diccio[input]);
               }
               else
               {
                   Console.WriteLine("Not found");
               }
           }

          // Console.ReadLine();

    }
}
```

Line: 29 Col: 21

Upload Code as File Test against custom input Run Code Submit Code

Congrats, you solved this challenge!

- ✔ Test Case #0
- ✔ Test Case #1
- ✔ Test Case #2
- ✔ Test Case #3
- ✔ Test Case #4

Next Challenge