

# Hashing

## Task 1

Given two arrays of integers, your task is to find whether array-2 is a subset of array-1 or not. Both arrays are unsorted and consist of distinct elements.

There will be two lines of input representing the arrays. Both lines end with -1 representing the end of that array.

**Output:**

- 'Yes': if array-2 is a subset of array-1
- 'No': if array-2 is not a subset of array-1

Sample Input	Sample Output
12 14 9 7 22 1 0 45 39 5 6 -1 7 5 6 -1	Yes
1 2 9 88 41 0 17 11 6 -1 100 52 2 8 -1	No

**Constraint:** Provide  $O(n)$  solution.

**Note:** Explore the 'unordered map' STL

([https://www.geeksforgeeks.org/unordered\\_map-in-cpp-stl/](https://www.geeksforgeeks.org/unordered_map-in-cpp-stl/) )

## Task 2

Given a sentence, you have to find the word(s) that occur more than once. Ignore the punctuation marks.

Input	Output
data atad structure atad data	data 2 atad 2
I know you know this, but you do not know of unknown trolls, because no known trolls will sew by windows, though they will owe you a hello when they throw a hoe, as it will go low and blow a hole in that window and so it will follow, that it happened awhile ago, as a troll will stand on a knoll and show you how to throw snow tomorrow at a rhino named Joe, who plays piano as he sips a cappucino and sings soprano in an inferno caused by a volcano in Reno with a casino at the bottom. Of the	volcano 2 a 9 by 2 will 5 at 2 throw 2 in 3 you 4 it 3

volcano.	the 2 know 3 and 4 trols 2 they 2 as 3 that 2
I know you know this but you do not know of unknown trols because no known trols will sew by windows though they will owe you a hello when they throw a hoe as it will go low and blow a hole in that window and so it will follow that it happened awhile ago as a troll will stand on a knoll and show you how to throw snow tomorrow at a rhino named Joe who plays piano as he sips a cappucino and sings soprano in an inferno caused by a volcano in Reno with a casino at the bottom. Of the volcano	volcano 2 a 9 by 2 will 5 at 2 throw 2 in 3 you 4 it 3 the 2 know 3 and 4 trols 2 they 2 as 3 that 2
This refers to an exam where James had written 'had had' where John had written just ' had'. The examiner had approved James' version.	James 2 where 2 had 6 written 2
This refers to an exam where James had written had had where John had written just had The examiner had approved James version	James 2 where 2 had 6 written 2

**Note:**

- Show the words in any order.
- May use `getline(cin, sentence)` to take the input.

## Task 3

### Problem Statement

You have just moved from Waterloo to a big city. The people here speak an incomprehensible dialect of a foreign language. Fortunately, you have a dictionary to help you understand them.

### Input

Input consists of several dictionary entries, followed by a blank line, followed by a message consisting of many words. Each dictionary entry is a line containing an English word, followed by a space and a foreign language word. No foreign word

appears more than once in the dictionary. The message is a sequence of words in the foreign language, one word on each line.

### Output

Output is the message translated to English, one word per line. Foreign words not in the dictionary should be translated as 'eh'.

### Sample Test Case(s)

Input	Output
dog ogday cat atcay pig igpay froot ootfray loops oopslay  atcay ittenkay oopslay	cat eh loops

## Task 4

Refer to Task 1. Here, instead of finding whether array 2 is only a subset of array 1, we consider whether array 2 is an ordered subset of array 1. In this context, we define an array as an ordered subset of another original array if its elements maintain their relative order from the original array but may not necessarily be contiguous.

Like before, there will be two lines of input representing the arrays. Both lines end with -1 representing the end of that array.

### Output:

- 'Yes': if array 2 is an ordered subset of array 1
- 'No': if array 2 is not an ordered subset of array 1

Sample Input	Sample Output
12 14 9 7 22 1 0 45 39 5 6 -1 7 5 6 -1	Yes
12 14 9 7 22 1 0 45 39 5 6 -1 7 6 5 -1	No
1 2 9 88 41 0 17 11 6 -1 100 52 2 8 -1	No

**Constraint:** Provide  $O(n)$  solution.