

Lab: Associative arrays

Problems for lab for the ["PHP Fundamentals" course @ SoftUni](#).

You can check your solutions in [Judge](#).

1. Counting characters in text

Write a program that reads a text and counts the occurrences of each character in it. Print the results.

Examples

Input	Output
aaabbaaabbcc	a -> 6 b -> 5 c -> 3

Hints

First we read the text

```
$text = readline();
```

Then we create array

```
$letters = [];
```

Iterate over the text:

```
for($i = 0; $i < strlen($text); $i++){  
    $char = $text[$i];
```

Check if the current char is set as key in the array (**key_exists()**), and if it is- add **one** to its value.

In other case- set the char as key, with initial value 1.

```
if (!key_exists($char, $letters))  
    $letters[$char] = 0;  
$letters[$char]++;
```

At the end print all key -> values from the array.

```
foreach($letters as $k => $v) {  
    echo "$k -> $v" . PHP_EOL;  
}
```

2. Count Real Numbers

Read a **list of real numbers** and **print them in ascending order** along with their **number of occurrences**.

Examples

Input	Output
8 2.5 2.5 8 2.5	2.5 -> 3 8 -> 2
1.5 5 1.5 3	1.5 -> 2 3 -> 1 5 -> 1
-2 0.33 0.33 2	-2 -> 1 0.33 -> 2 2 -> 1

Hints

Read the input and convert it to array:

```
$numbers = readline();  
$arr = explode( delimiter: " ", $numbers );
```

Initialize empty array

```
$result = [];
```

Check if the current number is set as key in the array (**key_exists()**), and if it is- add one to its value.
In other case- set the number as key, with initial value 1.

```
foreach ( $numbers as $num ) {  
    if ( !key_exists( $num, $numbersArr ) ) {  
        $numbersArr[ $num ] = 1;  
    } else {  
        $numbersArr[ $num ] ++;  
    }  
}
```

Sort:

```
krsort( &array: $result );
```

And print the array

```
foreach( $result as $key => $val ) {  
    echo $key . " -> " . $val . PHP_EOL;  
}
```

3. Odd Occurrences

Write a program that extracts from a given sequence of words all elements that present in it **odd number of times** (case-insensitive).

- Words are given in a single line, space separated.
- Print the result elements in lowercase, in their order of appearance.

Examples

Input	Output
Java C# PHP PHP JAVA C java	java c# c
3 5 5 hi pi HO Hi 5 ho 3 hi pi	5 hi
a a A SQL xx a xx a A a XX c	a sql xx c

Hints

- Use an **associative array** to count **the occurrences of each word (just like in the previous problem)**.
- Pass through all **key-value pairs** in the associative array and append to the results array all **keys** that have **odd value**.
- Print the results list.

4. Sum by Town

Read towns and incomes (**on a single line**) and print an array holding the total income for each town (see below). Print the towns in their natural order as object properties.

Examples

Input	Output
Sofia, 20, Varna, 10, Sofia, 5	Sofia => 25 Varna => 10
Plovdiv, 40, Pernik, 20, Vidin, 8, Sliven, 44, Plovdiv, 1, Vidin, 7, Chirpan, 0	Plovdiv => 41 Pernik => 20 Vidin => 15 Sliven => 44 Chirpan => 0

5. Word Synonyms

Write a program that keeps an associative array with synonyms. The **key** of the array will be the **word**. The **value** will be a **list of all the synonyms of that word**. You will be given a number **n**. On the next **2 * n** lines you will be given a **word** and a **synonym** each on a separate line like this:

- {word}
- {synonym}

If you get the same word twice just add the new synonym to the list.

Sort the words first by the count of synonyms (**values**) in descending order, and then by alphabetical order of the words (**keys**)

Output

Print the words in the following format:

"{word} - {synonym1, synonym2,..., synonymN}"

Examples

Input	Output
3 smart clever cute adorable cute charming	cute - adorable, charming smart - clever
5 task problem fashion model fashion look cute adorable task assignment	fashion - model, look task - problem, assignment cute - adorable

Hints

- Use **array as associative array** to keep track of all words

```
$arr = [];
```

- Read $n * 2$ lines
- Add the word in the associative array if it is not present

```
if (!key_exists($word, $arr)) {  
    $arr[$word] = [];  
}
```

- Add the synonym as value to the given word

```
array_push( &array: $arr[$word], $synonym );
```

- Sort :

```
uksort( &array: $arr, function ($key1, $key2) use ($arr) {  
    $countSynonyms1 = count($arr[$key1]);  
    $countSynonyms2 = count($arr[$key2]);  
    if ($countSynonyms1 == $countSynonyms2) {  
        return $key1 <=> $key2;  
    }  
    return $countSynonyms2 <=> $countSynonyms1;  
});
```