

As an avid user of CodeFights, you find it frustrating that there are no debugging and recovery tasks in your favorite language, PHP. You decide to help the platform by translating solutions in JavaScript into PHP.

You quickly discover that this approach is quite convenient: sometimes all you have to do is substitute the function arguments by adding the `$` sign in front of them. At first you do this manually, but soon realize that it won't get you far enough.

Now you need to implement a function that, given a `solution` written in JavaScript and its `args`, adds a `$` sign in front of each `args[i]` (unless there is already a dollar sign present).

Given a `solution` in JavaScript and its `args`, return the almost-PHP solution.

Example

For

```
solution =
  "function add($n, m) {\t
    return n + $m;\t
  }"
```

and `args = ["n", "m"]`, the output should be

```
programTranslation(solution, args) =
  "function add($n, $m) {\t
    return $n + $m;\t
  }"
```

Input/Output

- **[execution time limit] 4 seconds (js)**
- **[input] string solution**

Solution written in JavaScript. It is guaranteed that:

- the given code snippet is correct and can be executed in the CodeFights environment with `$` symbols removed;
- `solutions` does not contain comments or string variables;
- `solutions` does not start with one of the `args`.

Due to system limitations, tabulation (`\t`) characters are used instead of newlines (`\n`).

Guaranteed constraints:

`40 ≤ solution.length ≤ 200`.

- **[input] array.string args**

An array of distinct function arguments. It is guaranteed that each argument is valid, i.e. it consists only of uppercase and lowercase letters `'A'` through `'Z'`, the underscore `'_'` and, except for the first character, the digits `'0'` through `'9'`. It is also guaranteed that no argument coincides with one of the reserved words.

Guaranteed constraints:

`1 ≤ args.length ≤ 10`,
`1 ≤ args[i].length ≤ 10`.

- **[output] string**

The given `solution` with `args` replaced to PHP-style.

[JavaScript (ES6)] Syntax Tips

```
// Prints help message to the console
```