

Jet.com customers can easily find the item they are looking for by browsing by category. Categories are stored in a catalog that is updated on a regular basis as inventory changes. Your goal is to implement an algorithm that updates the catalog with new items.

The catalog is given as a two-dimensional array. For each  $i$ , `catalog[i][0]` represents the name of the corresponding category, and `catalog[i][j]` for  $j > 0$  is the name of some item within this category, which can also be the category of some other items. For each  $i$  all elements of `catalog[i]` except the first are sorted [lexicographically](#), and catalog arrays are sorted *lexicographically* by their first elements. The name of the topmost directory is "root".

Given a list of updates, update the catalog with the new items and return the result.

### Example

For

```
catalog = [
    ["Books", "Classics", "Fiction"],
    ["Electronics", "Cell Phones", "Computers", "Ultimate item"],
    ["Grocery", "Beverages", "Snacks"],
    ["Snacks", "Chocolate", "Sweets"],
    ["root", "Books", "Electronics", "Grocery"]]
```

and

```
updates = [
    ["Snacks", "Marmalade"],
    ["Fiction", "Harry Potter"],
    ["root", "T-shirts"],
    ["T-shirts", "CodeFights"]]
```

the output should be

```
catalogUpdate(catalog, updates) = [
    ["Books", "Classics", "Fiction"],
    ["Electronics", "Cell Phones",
    "Computers", "Ultimate item"],
    ["Fiction", "Harry Potter"],
    ["Grocery", "Beverages", "Snacks"],
    ["Snacks", "Chocolate", "Marmalade",
    "Sweets"],
    ["T-shirts", "CodeFights"],
    ["root", "Books", "Electronics",
    "Grocery", "T-shirts"]]
```

### Input/Output

- **[time limit] 4000ms (py)**
- **[input] array.array.string catalog**

The initial catalog in the format described above. It is guaranteed that the catalog is correct, i.e. there are no duplicate elements, `category[i][0] = "root"` for some  $i$ , and for each  $t \neq i$  `category[t][0]` is equal to exactly one `category[j][k]`, where  $j \neq t$  and  $k > 0$ .

*Constraints:*

$1 \leq \text{catalog.length} \leq 10$ ,

$2 \leq \text{catalog}[i].\text{length} \leq 5$ ,

$1 \leq \text{catalog}[i][j].\text{length} \leq 45$ .

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

Array of updates. Each update is a two-element array in the format [`<parent_category>`, `<item_name>`], where `<parent_category>` is the name of the category the item belongs to, and `<item_name>` is the item name.

The elements of update should be considered in the order they are given.

It is guaranteed that `<parent_category>` is always present in the catalog at the time the corresponding update is made.

*Constraints:*

$0 \leq \text{updates.length} \leq 15$ ,

$2 \leq \text{updates}[i].\text{length} \leq 2$ ,

$3 \leq \text{updates}[i][j].\text{length} \leq 24$ .

- **[output] array.array.string**

The updated catalog, formatted as it is initially given. The elements should be sorted as described above, and all elements of the resulting array should contain at least two elements each.