Arrays of arrays — The pizza restaurant

The objective of this work is to realize a (simplified) selection procedure in a restaurant menu.

An Italian restaurant opened his doors in September (well, he did not chose the best time of the year, that's for sure, but he hopes that he will still be able to work thanks to take away). In this subject, we care about the "Pizzas" section of the menu, depicted in Table 1.

Table 1: — Pizzas —

00	Marinara	(tomato, garlic, oregano, olive oil)
01	Regina	(tomato, mozzarella, ham, mushrooms)
02	Artica	(arugula, mozzarella, shrimps, salmon)
03	Margherita	(tomato, mozzarella, basil, olive oil)
04	Romana	(tomato, anchovy, oregano, olive oil)
05	Napoletana	(tomato, black olives, oregano, olive oil)
06	Siciliana	(tomato, basil, anchovy, capers)
07	Quattro stagioni	(ham, mushrooms, olives, artichokes)
08	Quattro formaggi	(mozzarella, Gorgonzola, Parmesan cheese, goat cheese)
09	Calzone	(mozzarella, ham, tomato, egg)
10	Guarnita	(tomato, pepperoni, green peppers, mushrooms)
11	Hawaiana	(mozzarella, tomato, ham, pineapple)

The restaurant owner remarked that his clients prefer some ingredients over others. He wishes to write a program that takes as input the name of an ingredient (then, multiple names) and that outputs the names of pizzas containing this (these) ingredient(s).

A pizza is encoded as an array of strings representing its ingredients. For example, the pizza "Regina" corresponds to the array:

```
["tomato", "mozzarella", "ham", "mushrooms"]
```

In this subject, all pizzas contain exactly 4 ingredients. The array "names" (it is an array of strings) and the array "ingredients" (it is an array of arrays of strings) written in Listing 1 can be downloaded from http://www.barsamian.am/2021-2022/S6ICTE/TP9_Pizzas.py.

```
names = ["Marinara", "Regina", "Artica", "Margherita", "Romana", "Napoletana",
         "Siciliana", "Quattro stagioni", "Quattro formaggi", "Calzone",
         "Guarnita", "Hawaiana"]

ingredients = [["tomato", "garlic", "oregano", "olive oil"],
               ["tomato", "mozzarella", "ham", "mushrooms"],
               ["arugula", "mozzarella", "shrimps", "salmon"],
               ["tomato", "mozzarella", "basil", "olive oil"],
               ["tomato", "anchovy", "oregano", "olive oil"],
               ["tomato", "black olives", "oregano", "olive oil"],
               ["tomato", "basil", "anchovy", "capers"],
               ["ham", "mushrooms", "olives", "artichokes"],
               ["mozzarella", "gorgonzola", "parmesan cheese", "goat cheese"],
               ["mozzarella", "ham", "tomato", "egg"],
               ["tomato", "pepperoni", "green peppers", "mushrooms"],
               ["mozzarella", "tomato", "ham", "pineapple"]]
```

Listing 1: Arrays describing the pizzas.

The pizzas are numbered from 0 to 11. For each i in $\{0, 1, \ldots, 11\}$, the i^{th} pizza has the name `names[i]` and contains the ingredients in the array `ingredients[i]`.

1 Study of the function "pizza_with"

We consider the function "pizza_with", in Figure 1, that takes as input an ingredient and is supposed to return, in an array of integers, the indexes of pizzas containing this ingredient.

Input:
ingredient is a string.

Variables:
i is an integer.
pizzas is an array of integers.

Instructions of the function:
1 *pizzas* ← [ ] (An empty array)
2 For i from 0 to 11
3 If $ingredients[i][0] = ingredient$, Then
4 $pizzas \leftarrow pizzas \cup [i]$ (We append the value i to the array — we add i at the end)
5 End If
6 End For
7 Return *pizzas*

Figure 1: Algorithm for the function "pizza_with".

1. What are the names of the pizzas that contain mozzarella? What does the function "pizza_with" return when we give to the input `ingredient` the value `"mozzarella"`?
2. There is thus something wrong with the function. Please write a function "pizza_with_bis" that is the corrected version of "pizza_with".

2 Intersection of two arrays

Write a function "common_elements", in natural language, that takes as input two arrays of integers, and that returns an array containing the elements common to both arrays.

- the call `common_elements([1, 2, 3, 5], [2, 5, 7, 8])` must return `[2, 5]`

Hint: for each element of the first array, we can check whether this element is in the second array (this check can be done thanks to a loop through the elements of the second array). If this is true, we know that this element is a common element.

3 Python

1. Implement the functions "pizza_with_bis" and "common_elements" from previous sections.

   Hint: to add a value i to an array a, you can write `a = a + [i]` (as we have seen in Work n°8) or `a.append(i)`.
2. Write a program that asks the user the name of an ingredient and that prints the names of all pizzas containing this ingredient.
3. Write a program that asks the user the name of two ingredients and that prints the names of all pizzas containing both ingredients.

4 Going further

http://www.barsamian.am/2021-2022/S6ICTE/TP9_Going_further.pdf.