

Fiche de TP numéro 8 - Dictionnaires

Le contenu de votre réfrigérateur et du placard de votre cuisine est mémorisé dans une structure du type `TIngredients = dico(string:int)` où la clé est le nom d'un produit alimentaire, et la valeur la quantité de ce produit.

Exemples :

```
# TIngredients : dico(str,int)
# Pour les exemples suivants, on utilisera :
ma_cuisine = {} # aujourd'hui, ma cuisine est vide
```

Exercice 1 : Spécifier puis écrire une fonction `ajoute_produit` qui permet d'ajouter un produit en une certaine quantité dans la cuisine de type `TIngredients`. Le contenu de la cuisine, le produit et la quantité sont passés en paramètres. Le contenu de la cuisine est modifié.

Exemples :

```
>>> ajoute_produit(ma_cuisine, 'chocolat', 50)
>>> ma_cuisine
{'chocolat': 50}
>>> ajoute_produit(ma_cuisine, 'margarine', 250)
>>> ajoute_produit(ma_cuisine, 'oeufs', 6)
>>> ma_cuisine
{'chocolat': 50, 'margarine': 250, 'oeufs': 6}
>>> ajoute_produit(ma_cuisine, 'chocolat', 100)
>>> ma_cuisine
{'chocolat': 150, 'margarine': 250, 'oeufs': 6}
```

Exercice 2 : Spécifier puis écrire une fonction `produits_present` qui teste si tous les produits d'une liste sont présents dans la cuisine.

Exemples :

```
>>> produits_present(ma_cuisine, ['chocolat','oeufs'])
True
>>> produits_present(ma_cuisine, ['chocolat','oeufs','beurre','sucre'])
False
```

Exercice 3 : Spécifier puis écrire une fonction `present_en_qte` qui teste si un produit est présent en quantité suffisante dans une cuisine.

Exemples :

```
>>> present_en_qte(ma_cuisine, 'beurre', 100)
False
>>> present_en_qte(ma_cuisine, 'chocolat', 100)
True
>>> present_en_qte(ma_cuisine, 'oeufs', 8)
False
```

Exercice 4 : Spécifier puis écrire une fonction `ranger_les_courses` qui ajoute un panier rempli de courses à la cuisine. Le panier et la cuisine sont fournis en paramètres.

Exemples :

```
>>> mes_courses = {}
>>> ajoute_produit(mes_courses, 'bananes', 5)
>>> ajoute_produit(mes_courses, 'sucre', 800)
>>> ajoute_produit(mes_courses, 'farine', 1.5)
>>> ma_cuisine
{'margarine': 250, 'chocolat': 150, 'oeufs': 6}
>>> mes_courses
{'bananes': 5, 'sucre': 800, 'farine': 1500}
>>> ranger_les_courses(mes_courses, ma_cuisine)
>>> ma_cuisine
{'margarine': 250, 'chocolat': 150, 'oeufs': 6, 'bananes': 5,
 'sucre': 800, 'farine': 1500}
```

Une recette de cuisine est mémorisée dans une structure de même type (on ne s'intéresse ici qu'aux ingrédients nécessaires à la réalisation d'une recette). Une collection de recettes est une structure du type `TRecettes = dico(string:TIngredients)` où la clé est le titre d'une recette.

Exemples :

```
# TRecettes : dico(str,TIngredients)
# On supposera les variables suivantes définies pour les exemples
>>> gateau_choc = {'oeufs':4,'sucre':150,'farine':80,\
                  'beurre':200,'chocolat':200}
>>> quatre_quarts = {'oeufs':4,'sucre':250,'farine':250,'beurre':250}
>>> les_recettes = {'omelette' :{ 'oeufs': 4, 'lait (en cl)': 5},\
                  'soupe' : {'poireau': 4, 'pommes de terre': 2},\
                  'fondant au chocolat' : gateau_choc,\
                  'quatre-quarts' : quatre_quarts}
```

Exercice 5 : Spécifier puis écrire la fonction `recette_possible(cuisine,une_recette)` qui prend en paramètre deux éléments de type `TIngredients` et qui retourne vrai si la recette peut être faite à partir du contenu du réfrigérateur.

```
>>> recette_possible(ma_cuisine, les_recettes['soupe'])
False
>>> recette_possible(ma_cuisine, gateau_choc)
False
```

Exercice 6 : Spécifier puis écrire la fonction `les_recettes_possibles(cuisine,les_recettes)` qui retourne la liste des noms des recettes qui peuvent être faites à partir du contenu du réfrigérateur. Chaque recette de la liste retournée doit pouvoir être faite, mais on ne pourrait pas forcément faire en même temps toutes les recettes de la liste.

```
>>> les_recettes_possibles({'oeufs':4, 'sucre':350, 'farine':300,
 'chocolat':200, 'beurre':250}, les_recettes)
['fondant au chocolat','quatre-quarts']
```

Exercice 7 : Spécifier puis écrire la fonction `courses_pour_recette(une_cuisine, une_recette)` qui, cette fois-ci, retourne les courses à faire pour pouvoir faire une recette.

```
>>> courses_pour_recette(ma_cuisine, gateau_choc)
{'beurre': 200, 'chocolat': 50}
>>> courses_pour_recette(ma_cuisine, les_recettes['soupe'])
{'poireau': 4, 'pommes de terre': 2}
```

Exercice 8 : Spécifier puis écrire la fonction `tous_les_ingredients` qui rassemble en un seul dictionnaire tous les ingrédients des recettes prévues pour le repas.

```
>>> tous_les_ingredients(les_recettes)
{'oeufs': 12, 'lait (en cl)': 5, 'poireau': 4, 'pommes de terre': 2,
 'sucre': 400, 'farine': 330, 'beurre': 450, 'chocolat': 200}
```

Exercice 9 : Spécifier puis écrire la fonction `toutes_les_courses(cuisine, les_recettes)` qui retourne la liste des courses nécessaires pour faire toutes les recettes de `les_recettes` (de type `TRecettes`).

```
>>> toutes_les_courses(ma_cuisine, les_recettes)
{'oeufs': 6, 'lait (en cl)': 5, 'poireau': 4, 'pommes de terre': 2,
 'beurre': 450, 'chocolat': 50}
```