

Mini DS2

mercredi 4 décembre

Le sujet comporte 4 pages.

Exercice 1. Cours

Q1. On dispose d'un dictionnaire `langages` dont les clés sont des noms de langages de programmation (type `str`), et la valeur associée est l'année de création de ce langage (type `int`). Une partie de ce dictionnaire est `langages = {'Fortran': 1954, 'SQL': 1974, 'C': 1972, ...}`.

Donner la commande permettant de :

a) Connaître le nombre d'éléments dans ce dictionnaire :

b) Obtenir l'année de création de Python et la stocker dans une variable `py` :

c) Rajouter dans ce dictionnaire le langage Go créé en 2009 :

Q2. Aurait-on pu regrouper les langages créés la même année en choisissant comme clés des listes de chaînes de caractères (par exemple une clé `['Ada', 'C++']` associée à la valeur `1983`) ? Justifier.

Q3. Compléter la fonction `minimum` ci-dessous qui prend en argument un dictionnaire dont les valeurs sont des flottants (`float`) et qui renvoie la valeur minimale du dictionnaire. Par exemple `minimum({'a': 5.0, 'b': 3.2, 'c': 4.7})` doit renvoyer `3.2`.

On pourra utiliser `float('inf')` pour créer un nombre correspondant à $+\infty$.

```
def minimum(dico: dict) -> float:  
    ...
```

Exercice 2. Retour sur le DS1

Q4. Écrire la commande permettant d'importer le module `random` et celle permettant d'importer **exclusivement** la fonction `cos` du module `math`.

Q5. Compléter la fonction `ligne` ci-dessous qui prend en argument un entier n et un flottant x , et qui renvoie une liste de taille n dont le j -ème élément vaut $\cos(x + j)$.

On commencera par une assertion vérifiant que n est strictement positif.

```
def ligne(n: int, x: float) -> [float]:  
    ...
```

Q6. En déduire une fonction `matrice` qui prend en argument un entier n , et qui renvoie une liste de n listes de taille n telles que le j -ème élément de la i -ème liste vaut $\cos(i + j)$.

Exercice 3. Cuisinons !

Dans tout cet exercice on choisit de représenter des recettes à l'aide de dictionnaires :

- les clés sont des chaînes de caractères (`str`) représentant les noms des ingrédients,
- les valeurs sont des entiers (`int`) correspondant à la quantité (toujours en grammes) demandée de cet ingrédient.

Q7. Écrire la ou les commande(s) permettant de créer un dictionnaire `smoothie` correspondant à la recette composée de 150 grammes de banane, 30 grammes de lait et 10 grammes de miel.

- Q8.** Compléter la fonction `masse` ci-dessous qui prend en argument un dictionnaire représentant une recette et qui renvoie un entier représentant la masse totale du résultat de la recette. Par exemple, `masse(smoothie)` doit renvoyer 190 (car $150 + 30 + 10 = 190$).

```
def masse(recette: dict) -> int:
    ...
```

Étant donné un stock d'ingrédients, on souhaite maintenant savoir si l'on peut réaliser une recette donnée. On choisit de représenter ledit stock par un dictionnaire ayant les mêmes caractéristiques que ceux pour les recettes. Par exemple si `stock = {miel: 100, lait: 1000, banane: 200}`, on peut réaliser la recette du smoothie décrite en **Q7**.

- Q9.** Compléter le tableau ci-dessous pour former un jeu de tests traitant les divers cas pouvant se produire :

Description cas	stock	recette	Sortie attendue
Tout ce qu'il faut	{'a': 1000, 'b': 100}	{'a': 30, 'b': 50}	True
⋮	⋮	⋮	⋮

- Q10.** Écrire une fonction `possible` qui prend en argument deux dictionnaires `stock` et `recette` et qui renvoie un booléen valant `True` si on peut faire la recette avec le stock donné, et `False` sinon.

Exercice 4. Premières requêtes SQL

On dispose d'une table **Voies** qui répertorie des voies d'escalade sur une falaise donnée. Elle est composée des attributs suivants (entre parenthèses on précise le type de chacun) :

- **nom** (TEXT), clé primaire ;
- **longueur** (INT), en mètres ;
- **cotation** (TEXT), donnant la difficulté de la voie (il s'agit d'un nombre suivi d'une lettre).

Dans tout cet exercice, on demande des requêtes dans le langage SQL.

Q11. Écrire une requête permettant d'obtenir tous les enregistrements de la table **Voies**.

Q12. Écrire une requête permettant d'obtenir le nom et la longueur de chaque voie de difficulté 6a.

Q13. Écrire une requête permettant d'obtenir les noms des voies de longueur supérieure ou égale à 20 mètres et dont la difficulté est 5c. Les résultats seront classés par ordre décroissant de longueur.