

Rattrapage de Programmation Numérique

Ibrahim ALAME

Mars 2024

Exercice 1

Que fait le programme suivant ? Que contiendra `res` à la fin du programme ?

```
l = [ 0,1,2,3,4,6,5,8,9,10]
res = True
for i in range (1,len(l)) :
    if l[i-1] > l[i] :
        res = False
```

Exercice 2

Ecrire une fonction qui calcule la somme des chiffres d'un entier positif.

Exercice 3

Que vaut n à la fin de l'exécution du programme suivant ? Expliquez en quelques mots le raisonnement suivi.

```
n = 0
for i in range (0,10) :
    if (n + i) % 3 == 0 :
        n += 1
print(n)
```

Exercice 4

Écrire un programme qui calcule l'élément u_{13} de la suite définie par :

$$\begin{cases} u_n = u_{n-1} + u_{n-2} \\ u_0 = 0, u_1 = 1 \end{cases}$$

u_{13} est le nombre de manières possibles pour une grenouille de monter en haut d'une échelle de 13 barreaux si elle peut faire des bonds de un ou deux barreaux seulement. Cette affirmation n'est pas à démontrer.

Exercice 5

On écrit le programme suivant :

```
li = range (0,10)
sup = [0,9]
for i in sup :
    del li[i]
print(li)
```

```
-----
TypeError                                Traceback (most recent call last)
<ipython-input-5-7a08c0bdf4c8> in <module>
      2 sup = [0,9]
      3 for i in sup :
----> 4     del li[i]
      5 print(li)

TypeError: 'range' object doesn't support item deletion
```

Que se passe-t-il ? Comment corriger le programme pour qu'il marche correctement ?
Qu'afficherait-il alors ?

Exercice 6

1. Quel est le résultat affiché par le programme suivant :

```
def lettre (s) :
    ens = []
    for i in range (0, len(s)) :
        c = s[i]
        if c in ens :
            ens.append(c)
    return ens
print(lettre("baaa"))
```

Est-ce que ce résultat change si on appelle la fonction ensemble_lettre avec un autre mot ?

2. Le programme précédent n'est vraisemblablement pas fidèle aux intentions de son auteur. Celui-ci avait pour objectif de déterminer l'ensemble des lettres différentes de la chaîne de caractères passée en entrée. Que faut-il faire pour le corriger ? Que sera le résultat de l'instruction `lettre("baaa")` en tenant compte de la modification suggérée ?

Exercice 7

Soit dico un dictionnaire où les clés sont des chaînes de caractères et les valeurs sont des listes de chaînes de caractères. Qu'affiche le code suivant:

```
# Définition d'un dictionnaire en plusieurs étapes:
animals = {'a': ['horse'], 'b': ['baboon'], 'c': ['giraffe']}
animals['d'] = ['donkey']
```

```
animals['d'].append('dog')
animals['d'].append('dingo')
# Affichage
print(animals)
```

1. Écrire une fonction `how_many (dic)` qui renvoie la somme du nombre de valeurs associées à tout clés du dictionnaire. Dans l'exemple ci-dessus, la fonction doit renvoyer 6.
2. Écrire une fonction `biggest(dic)` qui retourne la clé correspondant à l'entrée avec la plus grande nombre de valeurs qui lui sont associées. S'il y a plus d'une entrée de ce type, renvoyez l'une des clés correspondantes. Sur l'exemple ci-dessus, la fonction doit retourner 'd'.
3. Écrire une fonction `dstats(dic)` qui retourne un 2-tuple constitué de la somme du nombre de valeurs dans le dictionnaire et le plus grand nombre de valeurs. Sur l'exemple ci-dessus, la fonction devrait retourner le tuple (6, 3).

Exercice 8

Écrire une fonction récursive `issubset (st1, st2)` qui prend deux arguments de chaînes de caractères et retourne True si tous les caractères de la première chaîne `st1` apparaissent quelque part dans `st2`, et False sinon.

Indication:

- Si `st1` est vide alors `issubset('', st2)=True`.
- Si le premier caractère de `st1` est dans `st2` on a l'équivalence:

$$\text{issubset}(st1, st2) \iff \text{issubset}(st1[1:], st2)$$
- Sinon `issubset(st1, st2)=False`