

INTERROGATION N° 2

Jeudi 22 janvier 2026 (13h – 14h)

1

Un tri au choix

Écrire une fonction `tri` prenant en argument une liste d'entiers `t` non vide et qui trie en place celle-ci. On laisse le choix de l'algorithme utilisé et on rappelle que cette fonction ne doit rien renvoyer mais modifier son paramètre `t` d'entrée par permutations successives.

2

Suites de Collatz

On considère une suite d'entiers définie par $u_0 = x \in \mathbb{N}$ et

$$\forall n \in \mathbb{N}, u_{n+1} = \begin{cases} \frac{u_n}{2} & \text{si } u_n \text{ est pair} \\ 3u_n + 1 & \text{sinon} \end{cases}$$

Écrire une fonction récursive `collatz` prenant en argument des entiers `x` et `n` et renvoyant le terme u_n .

3

Listes alternées

Une liste $[t_0, \dots, t_{n-1}]$ d'éléments de $\mathbb{Z}^*$ est dite alternée si $n = 1$ ou ($n \geq 2$ et $\forall i \in \llbracket 0, n-2 \rrbracket, t_i t_{i+1} < 0$.)

Écrire une fonction `estAlternee` récursive prenant en argument une liste non vide d'entiers relatifs non nuls et renvoyant `True` si elle est alternée et `False` sinon.

4

Compression

Écrire une fonction récursive `compression` prenant en argument une liste `t` d'entiers et qui renvoie la liste obtenue en remplaçant toute suite d'éléments égaux consécutifs de `t` par un seul.

Par exemple, l'appel `compression(t)` pour $t = [1, 2, 2, 3, 3, 2, 4, 3, 3, 3]$ doit renvoyer $[1, 2, 3, 2, 4, 3]$.

5

Facteurs d'une liste

On appelle facteur(s) d'une liste $t = [t_0, t_1, \dots, t_{n-1}]$ toute liste de la forme

$$[t_i, t_{i+1}, \dots, t_j] \quad \text{avec } 0 \leq i \leq j \leq n-1$$

Par exemple, les facteurs de $[1, 2, 3]$ sont

$$[1], [2], [3], [1, 2], [2, 3], [1, 2, 3]$$

Écrire une fonction récursive `facteurs` prenant en argument une liste t d'entiers et qui renvoie la liste de tous les facteurs de t .

Par exemple, l'appel `facteurs(t)` pour $t=[1, 2]$ doit renvoyer $[[1], [2], [1, 2]]$ (à permutation près).

6

La fonction mystère

On considère la fonction suivante :

```
def mystere(t):
    if len(t)==0:
        return 1
    else:
        s=0
        for i in range(len(t)):
            tt=t[:]
            del tt[i]
            s=s+mystere(tt)
        return s
```

Copie de la liste t
Suppression de tt[i]

Que renvoie l'appel `mystere(t)` pour une liste d'entiers naturels t éventuellement vide ?

Un bonus sera accordé à toute démonstration du résultat par récurrence par la longueur de t .