

Séance 11b: TP NOTÉ (GROUPE 2)

Université Paris-Diderot

Instructions :

- **Pensez bien à tester votre code.**
- Toutes les fonctions doivent être dans **un seul fichier**.
- À la fin, il faut m'envoyer votre fichier source à adrien.koutsos@lsv.fr.

Exercice 1 (`forallContains`, ★)

Implémentez une fonction `forallContains` respectant la spécification suivante :

Spécification:

- *Entrée* : Une liste de listes d'entiers `lis`.
- *Sortie* : Un booléen assurant que toutes les lignes de `lis` contiennent chacune au moins deux occurrences de 6.

□

Exercice 2 (`piSquareOverSix`, ★)

Implémentez une fonction `piSquareOverSix` respectant la spécification suivante :

Spécification:

- *Entrée* : un entier `n`.
- *Sortie* : le flottant $\sum_{1 \leq k \leq n} \frac{1}{k^2}$, c'est à dire la somme de l'inverse du carré des entiers allant de 1 à `n` inclus.

□

Exercice 3 (`sameMultiple`, ★)

Implémentez une fonction `sameMultiple` respectant la spécification suivante :

Spécification:

- *Entrée* : deux listes d'entiers `l1` et `l2`.
- *Sortie* : une liste d'entiers `lis` dont la valeur en la coordonnée `i` vaut 1 si les deux listes `l1` et `l2` sont des multiples de 5, et -1 sinon. Si les tailles des listes sont différentes la fonction renvoie la liste `[0]`.

□

Exercice 4 (`printN`, ★)

Implémentez une fonction `printN` en utilisant une variable global `compteur` valant 0 avant le premier appel, et respectant la spécification suivante :

Spécification:

- *Entrée* : aucune.

- *Sortie* : affiche compteur fois une ligne contenant tata.
- *Effet* : incrémente compteur de 1.

Le première fois que l'on appelle printN elle n'affiche rien, la deuxième fois, elle affiche 'tata', la troisième fois elle affiche 'tata' deux fois etc. □

Exercice 5 (iterList, ★)

Implémentez une fonction iterList respectant la spécification suivante :

Spécification:

- *Entrée* : une liste d'entiers lis.
- *Sortie* : une liste de listes d'entiers res de même longueur que lis, et telle que si la position i de lis contient l'entier k , alors la position i de res contient une liste de 1 de longueur k .

Par exemple, iterList([2,4,1]) doit retourner la liste de listes d'entiers [[1,1], [1,1,1,1], [1]]. □

Exercice 6 (splitLetters, ★)

Implémentez une fonction splitLetters respectant la spécification suivante :

Spécification:

- *Entrée* : une chaîne de caractères m.
- *Sortie* : une liste de listes de caractères lis de longueur 2 telle que lis[0] contienne les lettres ayant une position paire dans m (dans l'ordre), et lis[1] contienne les lettres une position impaire dans m (dans l'ordre).

Par exemple, splitLetters('wikipedia') doit retourner [['i','i'],'e','i'], ['w','k'],'p','d','a']. □

Exercice 7 (syracuse, ★)

La suite de Syracuse partant de N_{ini} est définie par récurrence de la manière suivante :

$$u_0 = N_{ini}$$

$$\forall n \geq 0, u_{n+1} = \begin{cases} \frac{u_n}{2} & \text{si } u_n \text{ est pair.} \\ 3.u_n + 1 & \text{sinon} \end{cases}$$

Implémentez une fonction syracuse respectant la spécification suivante :

Spécification:

- *Entrée* : un entier N_{ini} et un entier n .
- *Sortie* : une liste contenant les n premiers termes de la suite de Syracuse partant de N_{ini} .

□