

Séance 5: LISTES ET PILES

L1 – Université Nice Sophia Antipolis

Exercice 1 (Occurrence, *)

1. Sans utiliser le mot-clé `in`, définissez une fonction `apparaît(x, L)` qui renvoie `True` si le nombre `x` apparaît dans la liste `L`.
2. Comment auriez-vous fait avec le mot-clé `in` ?
3. Définissez la fonction `contient(L1, L2)` qui renvoie `True` si tous les nombres de la liste de nombres `L1` apparaissent dans `L2`.
4. Définissez la fonction `commun(L1, L2)` qui renvoie le premier nombre de `L1` qui apparaît aussi dans `L2`. Si un tel nombre n'existe pas, `commun(L1, L2)` renvoie `False`.

□

Exercice 2 (Produit scalaire, *)

Dans cet exercice, on identifie les tuples avec les coordonnées dans \mathbf{R}^n muni de sa base orthonormale canonique.

1. Définissez une fonction `prod_scal(v1, v2)` qui prend en argument deux vecteurs représentés par des tuples de nombres de même longueur et qui renvoie leur produit scalaire. Par exemple, `prod_scal((1, 2, 3), (4, 5, 6))` renvoie `32 (= 4 + 10 + 18)`.
2. Déduisez-en une fonction `norme(v)` qui prend en argument un vecteur représenté par un tuple de nombres et qui renvoie sa norme. Par exemple, `norme((3,4))` renvoie `5.0`.

□

Exercice 3 (Produit externe, *)

Définissez une fonction `prod_ext(k, L)` qui prend en argument un nombre `k` et un vecteur \vec{v} représenté par une liste de nombres, et qui renvoie la liste représentant le vecteur $k \cdot \vec{v}$, sans modifier `L`. Proposez une première solution avec la méthode `append` et une autre solution avec une liste définie par compréhension. □

Exercice 4 (Substitution, *)

Définissez une procédure `remplace(x, y, L)` sans valeur de retour qui a pour effet de remplacer dans `L` tous les `x` par `y`. Par exemple, on aura au toplevel

```
1 >>> L = [1, 2, 4, 1, 3]
2 >>> remplace(1, 5, L)
3 >>> L
4 [5, 2, 4, 5, 3]
```

□

Exercice 5 (Reconnaître une permutation, **)

Définissez la fonction `est_permutation(L)` qui prend en argument une liste `L` de n entiers et qui renvoie `True` si `L` contient une et une seule fois tous les nombres de 0 à $n - 1$. Par exemple, `est_permutation([0,2,1])` renvoie `True` mais `est_permutation([0,1,0])` et `est_permutation([1,3,2])` renvoient `False`. Cherchez d'abord une solution quelconque, puis ajoutez la contrainte de parcourir une seule fois `L` (complexité linéaire).-

□

Exercice 6 (Crible d'Ératosthène, **)

Écrivez une fonction `eratosthene(n)` qui renvoie la liste de tous les nombres premiers inférieurs ou égaux `n` en appliquant l'algorithme du crible d'Ératosthène (demandez au chargé de TP ou à [Wikipédia](#) de vous expliquer l'algorithme). □

Exercice 7 (Les neurones de la lecture, **→* **)

1. Définissez une fonction `melange_mot(m)` qui renvoie un mot obtenu à partir de `m` en changeant aléatoirement l'ordre des lettres du mot `m` à l'exception de la première et de la dernière lettre. Par exemple, `melange_mot('cerise')` pourra renvoyer `'creise'` ou `'cierse'`.
Indication : Vous pourrez utiliser `list` pour convertir une chaîne de caractères en liste et la méthode (de la classe `string`) `join` pour faire l'opération inverse. La fonction `random.shuffle` permet de mélanger les éléments d'une liste.
2. Définissez une fonction `liste_des_mots('m')` qui prend en argument un texte composé de mots et de ponctuation (espace, deux points, virgule, point, point virgule, point d'exclamation, point d'interrogation, apostrophe) et qui renvoie la liste des mots et ponctuations qui constituent ce texte. Par exemple, `liste_des_mots('Hello, world!')` renverra `['Hello', ', ', ' ', 'world', '!']`
3. Définissez une fonction `melange_texte(s)` qui renvoie ce même texte où chaque mot a été mélangé comme à la question 1.
4. Mélangez l'article 1 de la déclaration universelle des droits de l'homme. Confirmez-vous ce que rapportent certains sites webs sur la capacité du cerveau humain à remettre dans le bon ordre les lettres en lecture rapide?

□

Exercice 8 (Dessin à la souris réversible, **)

Allez voir dans le cours sur *Processing* l'exemple du dessin à la souris. Modifiez-le pour que lorsqu'on appuie sur la flèche gauche on puisse annuler le dernier trait, puis l'avant-dernier, etc.

Indication : vous pourrez inclure dans votre monde (en plus de la position) la liste (pile) des traits à tracer et celle des traits annulés. □