
MPSI - ITC - TD 4 : LISTES

Ce sujet comporte 4 exercices sur 2 pages. L'exercice 4 est facultatif.

Pour les questions qui demandent à écrire une fonction, penser à documenter et tester vos fonctions.

Exercice 1 (Échauffement listes)

On rappelle qu'un appel de la forme `L.append(x)` modifie la liste `L` sans rien renvoyer, il serait donc absurde d'écrire `L=L.append(x)`.

1. Afficher la liste contenant les entiers de 1 à 10. Faire de même avec les entiers de 1 à 1000
2. Écrire une fonction `pairs` qui prend en entrée un entier n et renvoie la liste constituée des entiers pairs de 2 à n .
3. Écrire une procédure `affiche_liste` qui prend en entrée une liste et affiche tous ses éléments.
4. Écrire une procédure `modifier_sans_erreur` qui prend en entrée une liste et un entier i et double la valeur du i ème élément de L (attention aux indices), puis renvoie L . Attention, il est attendu de renvoyer L sans modification si i n'est pas un indice atteignable dans la liste L .

Exercice 2 (Tri comptage)

1. Écrire une fonction `nb_occurrences(l, x)` qui prend deux arguments `l` et `x` et qui renvoie l'entier correspondant au nombre de fois que `x` apparaît dans la liste `l`.
2. Écrire une fonction `tri_comptage(l)` qui prend en argument une liste ne contenant que des entiers entre 1 et 5, et qui utilise la fonction de la question 1 pour renvoyer une version triée de la liste `l`.

Exercice 3 (Palindrome)

Un palindrome est une suite de lettres qui se lisent indifféremment de gauche à droite ou de droite à gauche, par exemple : “radar”, “été”, “rotor”, “ressasser”, “Laval”, “engage le jeu que je le gagne” (Alain Damasio). On peut étendre la notion aux entiers, par exemple 1246421 est palindromique. Dans cet exercice on étend ce concept aux listes : on dira qu’une liste est palindromique si la suite de ses éléments est la même qu’on les regarde de gauche à droite (par indices croissants) ou de droite à gauche (par indices décroissants).

1. Créer une fonction `retourne` ayant comme argument une liste et qui renvoie la liste "retournée".
2. Écrire une fonction `est_palindrome` qui prend une liste en argument et renvoie un booléen qui indique si la liste est palindromique, en utilisant la fonction `retourne` de la question 1.
3. Pensez-vous à une façon de coder la question de l’exercice 2 sans créer une version retournée de la liste ?
4. (Facultatif) Faites en sorte que votre fonction puisse également prendre en argument une chaîne de caractères ou un entier.

Exercice 4 (Dérivation (facultatif))

On considère une fonction réelle f dérivable en a et on se propose d’estimer sa dérivée en a par

$$f'(a) \approx \frac{f(a+h) - f(a-h)}{2h}$$

1. Écrire une fonction `derive` de paramètres f , a et h qui retourne une valeur approchée de $f'(a)$. Tester cette fonction dans des cas connus avec différentes valeurs de h (par exemple, la fonction carré, la fonction exponentielle, etc ...).
2. Écrire une fonction `tangente` de paramètres f , a et h qui retourne un tuple de valeurs (m, p) où l’équation approchée de la tangente au graphe de f en a est de la forme $y = mx + p$. Cette fonction **doit** appeler la fonction `derive`.
3. Écrire une fonction `derive_compose` de paramètres f , g , a et h qui retourne une valeur approchée de $(g \circ f)'(a)$. Cette fonction **doit** appeler la fonction `derive`.