

L2 MIASHS – Informatique S5

TD1 : Révisions

2019-2020

Exercice 1

Que fait le programme suivant ? Expliquez les affichages.

```
1 def impaires(l):
2     m = []
3     for x in l :
4         if x%2==1 :
5             m.append(x)
6     return m
7
8 def doublePaires(l) :
9     for i in range(len(l)) :
10        if l[i]%2==0 :
11            l[i]=l[i]*2
12
13 listeA = []
14 for i in range(3):
15     listeA.append(i)
16 for i in range(3,6):
17     listeA.insert(0,i)
18 print("listeA =", listeA)
19
20 listeA[0] = listeA[0]//2
21
22 print(impaires(listeA))
23 doublePaires(listeA)
24 print("listeA =", listeA)
25
26 for i,x in enumerate(listeA) :
27     print(i+x)
28
29 listeB = [x**2 for x in range(0,7)]
30 print("listeB =", listeB)
31 listeC = [x for x in listeB if x not in [x for x in range(0,50) if x%2==0]]
32 print("listeC =", listeC)
```

Pour chacun des exercices, on vous demande de commencer par poser l'algorithme en langage naturel avant de coder.

On rappelle également qu'il est nécessaire de tester chaque fonction pour savoir si elle répond à ce qui est demandé. Pour chaque fonction, on réfléchira quelles valeurs de test sont pertinentes pour juger de son comportement.

Exercice 2

Ecrivez une fonction qui prend en entrée une liste de nombres et renvoie la sous-liste de ses éléments compris entre -1 et 1.

Par exemple si $x = [0.2, -0.5, 2, -0.2, -1.3]$, elle renvoie $[0.2, -0.5, -0.2]$

Exercice 3

Produisez puis affichez la liste des valeurs $4x+z$ où x et z varient chacun entre 0 inclus et 5 exclu.

Le résultat attendu est donc :

$[0, 1, 2, 3, 4, 4, 5, 6, 7, 8, 8, 9, 10, 11, 12, 12, 13, 14, 15, 16, 16, 17, 18, 19, 20]$

Exercice 4 – matrice

On représente souvent une matrice sous forme de liste de liste. Ainsi, $M = [[2, 3, 4], [7, 8, 9]]$ représente la matrice $\begin{pmatrix} 2 & 3 & 4 \\ 7 & 8 & 9 \end{pmatrix}$. Et $M[0][2] = 4$ affecte 4 à la case de coordonnées $(0, 2)$.

Produisez sous forme d'une liste de listes la matrice de taille 5×5 telle que la case qui est à la colonne x et à la ligne z vaut $4x+z$.

Le résultat attendu est donc :

$[[0, 1, 2, 3, 4], [4, 5, 6, 7, 8], [8, 9, 10, 11, 12], [12, 13, 14, 15, 16], [16, 17, 18, 19, 20]]$

Exercice 5 – palindrome

Un palindrome est un mot qui reste le même qu'on le lise de droite à gauche ou de gauche à droite. Ainsi les listes de chiffres $[1, 3, 2, 3, 1]$ ou $[5, 5, 1, 3, 3, 1, 5, 5]$ représentent des palindromes.

Question 1 Ecrivez une fonction qui teste si une liste représente un palindrome. Ecrivez deux versions, une sans utiliser le slicing, une avec.

Question 2 Ecrivez une fonction qui retourne la liste de tous les palindromes inclus dans une liste, en utilisant la fonction de la question 1.

Par exemple si $x = [5, 4, 3, 4, 4]$ elle retourne $[[5], [4], [4, 3, 4], [3], [4], [4, 4]]$.

Question 3 Même question, en une seule ligne, en utilisant une écriture en compréhension et du slicing.

Question 4 Ecrivez une fonction qui extrait d'une liste le plus grand palindrome qui s'y trouve. Si il y en a plusieurs de même longueur, elle en retourne un quelconque.

Par exemple si $x = [5, 4, 3, 4, 2, 1, 3, 3, 1, 5, 5]$ elle retourne $[1, 3, 3, 1]$.

Exercice 6 – souris monodimensionnelle

Vous êtes une petite souris située au point 0 dans un tunnel monodimensionnel. Ce tunnel contient des morceaux de fromage dont les positions sont indiquées par une liste. Programmez votre déplacement de façon à ce qu'à chaque étape :

1. vous vous dirigez vers le fromage le plus proche (mettre à jour votre position)
2. vous le mangez (mettre à jour la liste)

A la fin, votre algorithme doit renvoyer la liste des fromages dans l'ordre où ils sont mangés ainsi que la distance totale parcourue.

Par exemple, si la liste d'entrée est $[-6, -1, 2, 5]$, l'algorithme renvoie $[-1, 2, 5, -6]$ et une distance totale de $1 + 3 + 3 + 11 = 18$.

Exercice 7 – tri fusion

Pour trier une liste, on peut utiliser l'algorithme du *tri fusion*, qu'on peut définir ainsi :

- Si la liste est de longueur 0 ou 1, elle est triée.
- Si la liste est de longueur supérieure ou égale à deux, on procède ainsi :
 1. on coupe la liste en deux sous-listes de longueur égale (à 1 près),
 2. on trie chacune des sous-listes (en utilisant le même algorithme de tri fusion),
 3. on fusionne les deux sous-listes triées en une seule liste triée.

Question 1 Ecrivez une fonction qui prend deux listes triées et retourne la fusion des deux, triée. Par exemple, si les listes d'entrée sont $[-6, -1, 2, 5]$ et $[-3, 2, 3, 6, 7]$, l'algorithme renvoie $[-6, -3, -1, 2, 2, 3, 5, 6, 7]$.

Remarque : L'efficacité de l'algorithme de tri fusion repose sur le fait que cette fusion soit faite en temps linéaire.

Question 2 En utilisant la fonction précédente, écrivez une fonction qui retourne une liste triée contenant les mêmes valeurs que la liste passée en paramètre.