

Informatique en CPGE (2018-2019) Exercices sur les listes

Exercice 1

Déterminer ce qui sera affiché avec le code suivant :

```
a=[2,3,4,5,6,7,8,9][2:6][2]
print(a)
```

Exercice 2

Déterminer ce qui sera affiché avec le code suivant :

```
a=[]
for i in range(1,10):
    if i%2==0:
        a.append(i)
    else:
        a.append(2*i)
print(a)
```

Exercice 3

Ecrire une fonction qui prend une liste d'entiers en paramètre et renvoie deux nouvelles listes contenant respectivement les nombres pairs et les nombres impairs rangés dans le même ordre que dans la liste initiale.

Exercice 4

Ecrire une fonction qui prend en paramètre une liste de mots et renvoie deux nouvelles listes; la première liste contient les mots de moins de six lettres, la seconde contient les mots de six lettres ou plus.

Exercice 5

1. Ecrire une fonction **ordre1** qui prend en argument une liste de mots, ne renvoie rien et affiche les mots en fonction du nombre de lettres du plus petit au plus grand. Tester la fonction avec la liste :
maliste = ['toto', 'bonjour', 'a', 'oui', 'non'].
2. Reprendre la fonction **ordre1** et la modifier pour écrire une fonction **ordre2** qui cette fois n'affiche rien mais renvoie la liste des mots ordonnée en fonction du nombre de lettres de chaque mot.

Exercice 6

1. Donner la valeur des expressions suivantes :
(a) $[1, 2, 3] + [4, 5, 6]$

(b) $2 * [1, 2, 3]$

2. Ecrire une fonction `smul` à deux paramètres, un nombre et une liste de nombres, qui multiplie chaque élément de la liste par le nombre et renvoie une nouvelle liste.

Exemple : `smul(2, [1, 2, 3])` renvoie `[2, 4, 6]`.

3. Ecrire une fonction `vsom1` qui prend en paramètres deux listes de nombres de même longueur et qui renvoie une nouvelle liste constituée de la somme terme à terme de ces deux listes.

Exemple : `vsom([1, 2, 3], [4, 5, 6])` renvoie `[5, 7, 9]`.

4. Ecrire une fonction `vdif` qui prend en paramètres deux listes de nombres de même longueur et qui renvoie une nouvelle liste constituée de la différence terme à terme de ces deux listes (la première moins la deuxième).

Exemple : `vdif([1, 2, 3], [4, 5, 6])` renvoie `[-3, -3, -3]`.

5. Ecrire une fonction `vsom2` qui prend en paramètres deux listes de nombres de longueurs quelconques et qui renvoie une nouvelle liste constituée de la somme terme à terme de ces deux listes. Si une liste est plus courte que l'autre, les termes manquants sont remplacés par des 0.

Exemple : `vsom([1, 2, 3], [4, 5, 6, 7, 8])` renvoie `[5, 7, 9, 7, 8]`.

Exercice 7

1. Ecrire une fonction `f1` qui prend en paramètre une liste constituée de chiffres, les entiers 0 et 1, et renvoie une chaîne de caractères composée des caractères '0' et '1' dans le même ordre.

Par exemple `f1([1, 0, 1, 1, 0])` renvoie la chaîne '10110'.

2. Ecrire une fonction `f2` qui prend en paramètre une chaîne de caractères composée des caractères '0' et '1', et renvoie une liste composée des entiers 0 et 1 dans le même ordre.

Par exemple `f2('10110')` renvoie la liste `[1, 0, 1, 1, 0]`.

3. Ecrire une fonction `f3` qui prend en paramètre une liste constituée de chiffres, des 0 et des 1, et renvoie une liste de booléens obtenue de la manière suivante : un 0 est remplacé par `False` et un 1 est remplacé par `True`.

Par exemple `f3([1, 0, 1, 1, 0])` renvoie la liste `[True, False, True, True, False]`.

4. Ecrire une fonction `f4` qui prend en paramètre une liste constituée de booléens et renvoie une liste d'entiers obtenue de la manière suivante : `False` est remplacé par 0 et `True` est remplacé par 1.

Par exemple `f4([True, False, True, True, False])` renvoie la liste `[1, 0, 1, 1, 0]`.

5. Ecrire une fonction `decale_droite` qui prend en paramètre une liste constituée de chiffres, les entiers 0 et 1, et renvoie une nouvelle liste obtenue de la manière suivante : le dernier chiffre est supprimé, tous les autres sont décalés d'un rang vers la droite, un 0 est inséré à la première place.

Par exemple `decale_droite([1, 0, 1, 1, 0])` renvoie la liste `[0, 1, 0, 1, 1]`.