

Informatique PCSI

TP 9b : écriture d'un programme

Exercice 1

La fonction `moyennes` prend en paramètre une liste de nombres qui est non vide. Utiliser les deux commentaires pour corriger le code de cette fonction.

```
def moyennes(liste):
    n = len(liste)
    # initialisation d'une liste lisse avec la 1ère valeur de liste
    lisse = liste[0]
    for i in range(n-1):
        # moyenne de deux éléments consécutifs
        m = (liste[i] + liste[i+1] / 2)
        lisse.append(m)
    return lisse
```

Exercice 2

La fonction `lissage` prend en paramètre une liste de nombres qui est non vide. Utiliser le commentaire pour corriger le code de cette fonction.

```
def lissage(liste):
    n = len(liste)
    lisse = [liste[0]]
    for i in range(1, n-1):
        # ajoute à la liste la moyenne de trois éléments consécutifs
        lisse.append(sum(liste[i-1:i+1])/3)
    lisse.append(liste[n-1])
    return lisse
```

Il est habituel en Python, lorsque deux bornes sont précisées, que la borne gauche est incluse et la borne droite est exclue.

Exercice 3

La fonction qui suit prend en paramètre un entier `n` et doit renvoyer `True` si `n` est un nombre pair et `False` sinon.

```
def pair(n):
    """ n est un entier
    renvoie True si n est pair, sinon renvoie False"""
    if n % 2:
        return True
    else:
        return False
```

En utilisant cette fonction, on obtient exactement le résultat contraire à celui qui est prévu. Expliquer pourquoi et écrire une fonction qui satisfait à la spécification.

Exercice 4

Le programme qui suit contient deux fonctions `double1` et `double2`. Ces fonctions prennent une liste en paramètre et leur rôle doit être de concaténer la liste avec elle-même. Si une liste vaut `[1, 2]`, après l'exécution d'une fonction la liste doit avoir pour valeur `[1, 2, 1, 2]`.

```
def double1(liste):
    for i in range(len(liste)):
        liste.append(liste[i])

def double2(liste):
    i = 0
    while i < len(liste):
        liste.append(liste[i])
        i = i + 1
```

1. L'une des deux fonctions ne remplit pas son rôle et ne termine pas. Expliquer la non-terminaison.
2. Modifier le code de la fonction qui ne termine pas en gardant le même style de boucle afin qu'elle remplisse son rôle correctement.

Exercice 5

La fonction `f` prend en paramètre un nombre (de type `int` ou `float`).

```
def f(x):
    return 2 * x, 3 * x, 5 * x
```

Quel est le type du résultat renvoyé par la fonction `f`? Après l'instruction `a, b, c = f(2)`, quelle est la valeur de `c`?