

Informatique PCSI

TP 9b corrigé : écriture d'un programme

Exercice 1

Le premier commentaire précise que `lisse` est une liste. Or, la valeur affectée à `lisse` dans le code est un nombre. Donc l'initialisation doit être remplacée par `lisse = [liste[0]]`.

Le deuxième commentaire précise que `m` est la moyenne de deux termes consécutifs. Il y a donc une erreur dans le parenthésage. Écrire `m = (liste[i] + liste[i+1]) / 2`.

Exercice 2

L'expression `liste[i-1:i+1]` doit être une liste contenant trois éléments consécutifs. Or cette liste ne contient que deux éléments : `liste[i-1]` et `liste[i]`.

Il faut donc remplacer cette expression par `liste[i-1:i+2]`.

Exercice 3

Le test `if x % 2` n'est pas vérifié si `x` est pair puisque l'expression `x % 2` a pour valeur 0 dans ce cas, (c'est le reste de la division), et le nombre 0 est interprété comme `False`.

Pour obtenir un code correct, il suffit de permuter les valeurs renvoyées `True` et `False`. Une autre manière de corriger le code est de modifier le test en écrivant à la place `if x % 2 == 0`.

Exercice 4

On peut tester les fonctions sur machine.

1. Avec les deux fonctions, les listes sont modifiées (en place). La deuxième fonction pose problème car la longueur de la liste est réévaluée à chaque passage dans la boucle. La première fonction ne pose aucun problème.

```
>>> liste = [1, 2]
>>> double1(liste)
>>> liste
[1, 2, 1, 2]
```

Avec la fonction `double2`, le programme ne termine pas. On peut arrêter l'exécution en cours en appuyant simultanément sur les touches `ctrl` et `C`.

2. Pour que la fonction termine, il faut éviter de réévaluer la longueur de la liste à chaque passage dans la boucle. Pour cela, on l'évalue uniquement avant l'entrée dans la boucle.

```
def double3(liste):
    i = 0
    n = len(liste)
    while i < n:
        liste.append(liste[i])
        i = i + 1
```

Test de la nouvelle fonction :

```
>>> liste = [1, 2]
>>> double3(liste)
>>> liste
[1, 2, 1, 2]
```

Exercice 5

Le résultat renvoyé par la fonction f est de type tuple.

En effet, ce sont les virgules qui définissent un tuple. Les parenthèses peuvent être utilisées pour des questions de priorité ou de lisibilité.

Le résultat $f(2)$ est le tuple $(4, 6, 10)$, donc c prend la valeur 10.