

Corrigé sujet 03 - Année : 2022

Sujet 03 - 20222



1. Exercice 1

```
1 def delta(liste):
2     codage=[liste[0]]
3     for i in range(1,len(liste)):
4         codage.append(liste[i]-liste[i-1])
5     return codage
```

Commentaires

- On construit le codage en partant du premier élément de la liste. Les autres éléments sont les différences entre deux éléments consécutifs de la liste de départ.
- L'écriture de cette fonction peut aussi se faire (de façon plus concise) en utilisant les listes par compréhension :

Script Python

```
def delta(liste):
    return [liste[i]-liste[i-1] if i>0 else liste[i] for i in range(len(liste))]
```

2. Exercice 2

Bug

Le code fourni semble contenir une erreur, en effet, pour le compléter on utilise un `if True` à la ligne 23 ! On devrait donc soit se passer de la ligne 23, soit réécrire cette fonction.

```

1  class Noeud:
2      def __init__(self, g, v, d):
3          self.gauche = g
4          self.valeur = v
5          self.droit = d
6
7      def __str__(self):
8          return str(self.valeur)
9
10     def est_une_feuille(self):
11         """Renvoie True si et seulement si le noeud est une feuille"""
12         return self.gauche is None and self.droit is None
13
14
15
16     def expression_infixe(e):
17         s = "" # (1)
18         if e.gauche is not None: # (2)
19             s = '(' + s + expression_infixe(e.gauche)
20             s = s + str(e.valeur)
21         if e.droit is not None: # (3)
22             s = s + expression_infixe(e.droit) + ')'
23         if True : # (4)
24             return s

```

1. La variable `s` va contenir l'expression arithmétique
2. Si le noeud contient un fils gauche, on construit l'expression associée et on ajoute la valeur du noeud à la suite.
3. On construit la partie droite de l'expression (si elle existe)
4. Si on atteint cette ligne, l'expression a été construite en totalité, il reste à la renvoyer. Il ne devrait pas y avoir de `if` !

■ Commentaire

Sujet assez difficile *en plus d'être buggé* et qui utilise diverses notions du programme (arbre, récursivité) et qui présente de plus un aspect mathématique.