

DS d'informatique N°2 – corrigé

Exercice 1 : COMPRÉHENSION D'ALGORITHMES

1. — `truc(0)` renvoie 0;
— `truc(2)` renvoie $0 + 1 = 1$;
— `truc(5)` renvoie $0 + 1 + 2 + 3 + 4 = 10$.
2. `truc2([1,2,5,3,7,4,8,3,10], 3)` renvoie 5.
3. La fonction `truc2(L, s)` renvoie le nombre d'éléments de la liste `L` dont la valeur est strictement supérieure à `s`.

Enfin on s'intéresse à la fonction suivante :

```
def truc3(L, val):
    i = 0
    while i < len(L) and L[i] <= val :
        i += 1
        res = L[i]
    return [i,res]
```

4. `truc3([7,14,21,45,52,67,89,99],50)` renvoie `[4,52]` ?
5. La fonction `truc3(L, val)` renvoie une liste contenant l'indice et la valeur du premier élément de `L` dont la valeur est strictement supérieure à `val`.

Exercice 2 : ERREURS DE PROGRAMMATION

1. Cette erreur signifie que l'on a utilisé la variable `nb` avant de lui assigner une valeur. Il faudrait initialiser cette variable à 0 au début de la fonction.
2. Le problème est que le `return nb` se trouve à l'intérieur de la boucle `for`. La fonction retourne le résultat juste après la première itération et on n'obtient que les valeurs 0 et 1. Il faut réduire l'indentation de cette ligne pour l'aligner avec celle du `for`.
3. Il faut rajouter « : » à la fin de cette ligne pour corriger l'erreur de syntaxe.
4. Cette erreur signifie que l'on essaye d'accéder à un élément de la liste `L` en dehors de ses limites, en l'occurrence la dernière valeur de `i` est $n - 1$ avec n la taille de la liste `L`. Donc `L[i+1]` essaye d'accéder à l'élément numéro n de la liste `L` qui n'existe pas. Pour corriger l'erreur il faut écrire `for i in range(len(L)-1)`.
5. L'erreur se trouve aux lignes 5 et 6. Si les deux premiers éléments sont différents on renvoie `False` comme si la liste toute entière ne contenait pas de doublons. Il faut renvoyer `False` après être sorti de la boucle `for`. La fonction correcte est :

```
1 def contientDoublons(L):
2     for i in range(len(L) - 1)
3         if L[i] == L[i+1]:
4             return True
5     return False
```

Exercice 3 : QUELQUES FONCTIONS

1.

```
def moyenne(L):
    S = 0
    for v in L:
        S += v
    return S/len(L)
```

2.

```
def plusGrand(L):  
    pg = L[0]  
    for v in L:  
        if v > pg:  
            pg = v  
    return pg
```

3.

```
def listeCarres(n):  
    L = []  
    for i in range(n):  
        L.append((i+1)**2)  
    return L
```

4.

```
def estTrie(L):  
    for i in range(len(L)-1):  
        if L[i+1] < L[i]:  
            return False  
    return True
```