

DS d'informatique N°2

Exercice 1 : COMPRÉHENSION D'ALGORITHMES

On définit la fonction suivante :

```
def truc(n):
    res = 0
    for i in range(n):
        res = res+i
    return res
```

1. Donner les valeurs renvoyées par `truc(0)`, `truc(2)` et `truc(5)`.

On définit maintenant la fonction suivante :

```
def truc2(L, s):
    n = 0
    for v in L:
        if v > s:
            n = n + 1
    return n
```

2. Donner la valeur renvoyée par `truc2([1,2,5,3,7,4,8,3,10], 3)`.
3. Expliquer en quelques mots le but de la fonction `truc2`.

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. Que renvoie l'appel de la fonction `truc3([7,14,21,45,52,67,89,99],50)` ?
5. Expliquer en quelques mots ce que fait la fonction `truc3`.

Exercice 2 : ERREURS DE PROGRAMMATION

On souhaite écrire une fonction qui prend en paramètres une liste d'entiers `L` et un entier `n` et renvoie le nombre de fois où `n` apparaît dans la liste `L`.

```
1 def apparitions(L, n):
2     for i in range(len(L)):
3         if L[i] == n:
4             nb = nb + 1
5     return nb
```

Lorsqu'on appelle cette fonction, l'interpréteur Python renvoie l'erreur suivante :

```
File "<tmp 1>", line 4, in apparitions
    nb = nb + 1
```

`UnboundLocalError: local variable 'nb' referenced before assignment`

1. Expliquer ce que signifie cette erreur, et comment on peut la corriger.

On corrige cette erreur de syntaxe et on teste la fonction en effectuant les appels suivants :

- `apparitions([1,2,3,2,1,3,3,3], 1)` renvoie `1`;
- `apparitions([1,2,3,2,1,3,3,3], 2)` renvoie `0`;
- `apparitions([1,2,3,2,1,3,3,3], 3)` renvoie `0`;

Il y a clairement un problème avec ce programme, il ne renvoie pas la bonne valeur.

2. Expliquer où se trouve l'erreur et comment la corriger.

On souhaite maintenant écrire une fonction qui teste si une liste d'entiers que l'on suppose triée contient plusieurs fois le même nombre. On écrit la fonction suivante :

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

Lorsqu'on exécute le code qui définit cette fonction, Python refuse et affiche le message d'erreur suivant :

```
File "<tmp 1>", line 2  
    for i in range(len(L))  
        ~
```

SyntaxError: invalid syntax

3. Corriger l'erreur correspondante.

L'erreur corrigée, on appelle la fonction avec l'instruction `contientDoublons([1,2,3,4])`. L'interpréteur Python voit à nouveau rouge et affiche l'erreur suivante :

```
File "<tmp 1>", line 3, in contientDoublons  
    if L[i] == L[i+1]:  
IndexError: list index out of range
```

4. Que signifie cette erreur et comment la corriger ?

Enfin notre fonction s'exécute correctement, mais lors de tests, on obtient les résultats suivants :

```
— contientDoublons([1,2,3,4]) renvoie False;  
— contientDoublons([2,2,3,4]) renvoie True;  
— contientDoublons([1,2,2,3]) renvoie False;  
— contientDoublons([1,2,3,3]) renvoie False;
```

5. Visiblement notre fonction ne renvoie pas toujours le bon résultat. Expliquer où se trouve l'erreur et écrire une fonction `contientDoublons` correcte.

Exercice 3 : QUELQUES FONCTIONS

1. Écrire une fonction `moyenne(L)` qui prend en paramètre une liste `L` de nombres à virgule flottante et qui renvoie la moyenne de ces nombres.
2. Écrire une fonction `plusGrand(L)` qui prend en paramètre une liste `L` de nombres à virgule flottante et qui renvoie le plus grand de ces nombres.
3. Écrire une fonction `listeCarres(n)` qui prend en paramètre un nombre entier `n` et qui renvoie une liste contenant les carrés des nombres entiers compris entre 1 et `n`. Par exemple :
 - `listeCarres(3)` renvoie `[1, 4, 9]` ;
 - `listeCarres(5)` renvoie `[1, 4, 9, 16, 25]`.
4. (Un poil plus difficile) Écrire une fonction `estTrie(L)` qui prend en paramètre une liste `L` de nombres à virgule flottante et qui renvoie `True` si les éléments de cette liste sont triés par ordre croissant et `False` sinon.