

# Documentation et mise au point de programmes - EXERCICES

---

## Documenter son programme

---

### Exercice 1

Donnez un meilleur nom et une chaîne de documentation à la fonction suivante.

```
def f(a, b):  
    return a + b
```

### Exercice 2

Donnez un meilleur nom et une chaîne de documentation à la fonction suivante, le paramètre `t` étant un tableau.

```
def f(t):  
    s = 0  
    for i in range(len(t)):  
        s = s + t[i]  
    return s / len(t)
```

## Programmation défensive

---

### Exercice 3

On considère la fonction `indice_maxi_tab(T)` suivante. A l'aide de la construction `assert`, proposez un test vérifiant si la précondition sur le tableau `T` est validée (*on ne cherchera pas à écrire la fonction*).

```
def indice_maxi_tab(T):  
    """  
    Renvoie l'indice de la première occurrence de la valeur  
    maximale du tableau T. T est supposé non vide.  
    """  
    # TEST A ECRIRE ICI
```

## Exercice 4

On considère la fonction `quotient(a, b)` suivante. A l'aide de la construction `assert`, proposez un test vérifiant si les préconditions sont validées.

```
def quotient(a, b):  
    '''  
    Renvoie la valeur du quotient de a par b, b étant non nul.  
    '''  
  
    # TEST A ECRIRE ICI
```

## Tester ses programmes

---

### Exercice 5

1. En utilisant `assert`, donnez un jeu de tests de qualité pour la fonction suivante.
2. Faites ensuite passer vos tests à la fonction.

```
def multiplication(a, b):  
    '''  
    Renvoie le produit de a par b,  
    où a et b sont deux nombres quelconques.'''  
    return a * b
```

### Exercice 6

1. En utilisant `assert`, donnez un jeu de tests de qualité pour la fonction suivante.
2. Faites ensuite passer vos tests à la fonction.

```
def somme(t):  
    '''  
    Renvoie la somme des éléments du tableau t,  
    t étant un tableau de nombres  
    '''  
  
    s = 0  
    for i in range(len(t)):  
        s = s + t[i]  
    return s
```

### Exercice 7

On cherche à écrire une fonction `est_croissant(t)` qui renvoie `True` si le tableau `t` est trié dans l'ordre croissant et `False` sinon. En utilisant `assert`, donnez un jeu de tests de qualité pour cette fonction (*on ne*

cherchera pas à écrire le code de la fonction).

## Exercice 8

Un élève propose le code suivant pour la fonction `est_croissant(t)` de l'exercice précédent.

```
def est_croissant(t):  
    for i in range(len(t)):  
        if t[i+1] > t[i]:  
            return False  
    return True
```

1. Faites passer vos tests (de l'exercice précédent) à cette fonction.
2. Trouvez l'erreur en affichant l'état de certaines variables à des points stratégiques.
3. Corrigez alors le code de la fonction et vérifiez que tous les tests sont valides.

## Exercice 9

Un élève prétend que la fonction suivante teste l'appartenance de la valeur `v` au tableau `t`.

```
def appartient(v, t):  
    '''  
    Renvoie True sur v appartient à t, et False sinon.  
    '''  
    for i in range(len(t)):  
        if t[i] == v:  
            trouvee = True  
        else:  
            trouvee = False  
    return trouvee
```

1. Donnez des tests pour cette fonction, et en particulier des tests montrant plusieurs raisons pour laquelle cette fonction est incorrecte.
2. Vous corrigerez ensuite cette fonction.

## Exercice 10

On cherche à écrire une fonction `puissance(x, n)` dont la spécification est donnée dans la chaîne de documentation suivante.

```
def puissance(x, n):  
    '''  
    Renvoie la valeur de  $x^n$ , où  $n$  est un entier quelconque.  
    '''
```

1. Intégrez un jeu de tests bien choisi directement dans la chaîne de documentation.

2. Proposez ensuite un code pour cette fonction. Vous vérifiez que tous les tests sont valides et corrigerez la fonction si besoin.

**Références :**

- Documents ressources du DIU EIL Nantes, C. DECLERCQ.
- Numérique et Sciences Informatiques, 1re, T. BALABONSKI, S. CONCHON, J.-C. FILLIATRE, K. NGUYEN, éditions ELLIPSES : [Site du livre](#)

---

Germain BECKER & Sébastien POINT, Lycée Mounier, ANGERS

