

RÉCURSIVITÉ

Exercice 1 : Une fonction récursive dont le principe vous est familier

```
def recherche(x, deb, fin, L):
    if deb > fin:
        return "fini"
    t = (deb + fin) // 2
    if x == L[t]:
        return t
    elif x < L[t]:
        return recherche(x, deb, t - 1, L)
    else:
        return recherche(x, t + 1, fin, L)
```

Décrire sur papier les étapes lors de l'appel ci-dessous. Quel est le nom de l'algorithme utilisé ici ?

```
T = [2, 3, 4, 7, 11, 15, 17]
print recherche(4, 0, len(T) - 1, T)
```

Exercice 2 : Quand il y en a en trop...

```
def ed(C, M):
    if len(C) == 0:
        return M
    a = C[0]
    if a not in M:
        return ed(C[1:], M + a)
    else:
        return ed(C[1:], M)
```

Que renverra l'appel ci-dessous ?

```
ed("bcbfhhihcfghhi", "")
```

Exercice 3 : Comptage de caractères

Dans la fonction suivante, les appels récursifs sont à compléter :

Fonction `compte_a`

Entrée :

chaîne est une chaîne de caractères.

Sortie :

Renvoie le nombre de lettres 'a' contenues dans *chaîne*.

Corps de l'algorithme :

```
1 Si longueur(chaîne) == 0, alors
2     Renvoyer 0
3 Sinon
4     Si chaîne[0] == 'a', alors
5         ...
6     Sinon
7         ...
8     Fin Si
9 Fin Si
```

Exemples : `compte_a("blabla")` doit renvoyer 2, et `compte_a("dur")` doit renvoyer 0.

Exercice 4 : L'addition récursive

Dans la fonction suivante, le cas de base est à compléter :

Fonction plus

Entrées :

a et b sont deux entiers naturels.

Sortie :

Renvoie $a + b$.

Corps de l'algorithme :

```
1 Si  $b == 0$ , alors
2     ...
3 Sinon
4     Renvoyer  $plus(a + 1, b - 1)$ 
5 Fin Si
```

Remarques :

Le cas $b == 0$ est le "cas de base". C'est un cas pour lequel l'image à renvoyer ne nécessite pas d'appel à la fonction `plus`.

Dans les appels `plus(a + 1, b - 1)`, le second argument est décrémenté d'une unité à chaque appel et finira par être nul (b est un entier naturel dans l'appel initial). C'est ce qui garantit que le programme s'achèvera.

Exercice 5 : La fonction mystère

Fonction mystere

Entrées :

a et b sont deux entiers naturels.

Sortie :

A découvrir...

Corps de l'algorithme :

```
1 Si  $b == 1$ , alors
2     Renvoyer  $a$ 
3 Sinon
4     Renvoyer  $a + mystere(a, b - 1)$ 
5 Fin Si
```

1. Quel est le résultat renvoyé si on appelle `mystere(3, 5)` ?
2. Quel est le cas de base dans cette fonction récursive ?
3. Qu'est ce qui garantit dans les appels récursifs que le programme finira par s'arrêter ?
4. Que renvoie `mystere(a, b)` en règle générale (en fonction de a et b) ?