

Fiche d'exercices - Chapitre D.1 - La récursivité

Exercice 1

1.

Fonction récursive

```
else:  
    print("FIN")
```

2.

Fonction non récursive

3.

Fonction récursive

"else" implicite

4.

Fonction récursive

2 cas de bases

```
if n == 0:  
    return True
```

```
if n == 1:  
    return False
```

5.

Fonction récursive

2 cas de bases

```
if n == 0:  
    return False
```

```
if n == 1:  
    return True
```

Exercice 2

1.

Les valeurs prises par le paramètres n forment une suite d'entier strictement décroissants. On est donc sur d'atteindre le cas $n < 1$ (cas de base).

2.

La valeur absolue de $x - y$ diminue à chaque tour. Donc $x - y$ atteindra la valeur 0 donc $x = y$ (Le cas de base)
3.

Les valeurs prises par le paramètres x forment une suite d'entier strictement décroissants. On est donc sur d'atteindre le cas $x \leq 0$ (l'un des cas de base).

Exercice 3

```
def somme(n):  
    if n == 0:  
        return 0  
    else:  
        return somme(n-1)+n
```

Exercice 4

1. La fonction fait appelle a elle même. Elle est donc récursive.
2. (a) Si on exécute toujours le cas "False". On appel $A()$ à répétition. Donc on atteindra le maximum de la pile d'exécution. (MaximumRecursionError).

(b)

```
def A(n):  
    if n<=0 or choice([True, False]):  
        return "a"  
    else:  
        return "a" + A(n-1) + "a"
```

(c) Les valeurs de n forment une suite d'entiers positif strictement décroissante. On atteindra donc toujours le cas $n \leq 0$

3.

$B(0) \rightarrow$ "bab"

$B(1) \rightarrow$ "bab" + "bbabb"

$B(2) \rightarrow$ "bab" + "baaab" + "bbabb" + "bbbabbbb"

4.

(a)

```
def regleA(chaine):  
    n = len(chaine)  
    if n >= 2:  
        return chaine[0] == "a" and chaine[n-1] == "a" and regleA(raccourcir(chaine))  
    else:  
        return chaine == "a"
```