

Informatique PCSI
TP 5 : algorithmes récursifs

1 Énoncé des exercices

1.1 Exercice 1

Les seules opérations autorisées sur les nombres entiers sont soit ajouter 1, soit retrancher 1. La fonction `somme` renvoie la somme des entiers positifs `a` et `b`.

```
def somme(a, b):
    while b > 0:
        a = a + 1
        b = b - 1
    return a
```

Écrire une version récursive de cette fonction.

1.2 Exercice 2

En s'inspirant de l'exercice précédent, il est demandé d'écrire une fonction récursive qui renvoie la somme de deux entiers quelconques. Les seules opérations autorisées sur les nombres entiers sont ajouter ou retrancher 1.

Pour effectuer la somme $a + b$, distinguer les cas $b > 0$, $b < 0$ et $b = 0$.

1.3 Exercice 3

La fonction qui suit permet de savoir si un entier naturel est pair ou non.

```
def pair(n):
    while n > 0:
        n = n - 2
    return n == 0
```

Écrire une version récursive de cette fonction.

1.4 Exercice 4

On écrit en récursif une fonction étudiée au TP 4 qui calcule x^n .

Écrire une fonction récursive puissance qui prend en paramètres un flottant x non nul et un entier naturel n et qui renvoie x^n en se basant sur la définition mathématique : $x^0 = 1$ et $x^n = x \times x^{n-1}$ pour $n > 0$.

1.5 Exercice 5

Calcul de x^n

1. Écrire, comme à l'exercice précédent, une fonction récursive `puissance1` prenant en paramètres un flottant x non nul et un entier naturel n et qui renvoie x^n . Mais, comme dans le TP 4, appliquer la remarque suivante :
 $x^0 = 1$, si $n = 2k$ alors $x^n = x^{2k} = (x^k)^2$ et si $n = 2k + 1$ alors $x^n = x^{2k+1} = x(x^k)^2$.
2. Même question pour une fonction `puissance2`, mais en utilisant la remarque : $x^0 = 1$, si $n = 2k$ alors $x^n = x^{2k} = (x^2)^k$ et si $n = 2k + 1$ alors $x^n = x^{2k+1} = x(x^2)^k$.

1.6 Exercice 6

1. Écrire une fonction récursive `somme` qui prend en paramètre une liste de nombres et renvoie la somme des termes de cette liste.
2. Que penser du coût en temps et en espace d'une telle fonction ?
3. Expliquer avec l'exemple `somme ([4, 7, 2])` le déroulement de l'exécution de la fonction.