

## TD13 – Récursivité

---

Tous les exercices de ce TD doivent se faire exclusivement de manière récursive.

### Exercices sur les nombres

#### Exercice 1 :

Écrivez la fonction récursive qui calcule la somme des nb nombres entiers positifs ( $1 + 2 + 3 + \dots + nb$ ).

Écrivez également le programme principal qui teste cette fonction avec assert et écrivez tous les appels successifs qui se font quand on appelle la fonction avec nb=4.

#### Exercice 2 :

Écrivez la fonction récursive qui calcule **a** à la puissance **b**.

Écrivez également le programme principal qui teste cette fonction avec assert et écrivez tous les appels successifs qui se font quand on appelle la fonction avec : a=3 et b=4.

#### Exercice 3 :

Écrivez une fonction qui détermine si un nombre est pair.

#### Exercice 4 :

Écrivez une fonction qui calcule le pgcd de deux nombres strictement positifs a et b. Testez votre fonction avec des asserts.

#### Exercice 5 :

On se donne un entier n, par exemple n=2030, et on cherche le plus petit entier supérieur à n et qui se termine par X (X est forcément un nombre à 2 chiffres), par exemple X=42. Si n=2030 alors la réponse attendue est 2042.

Remarquer qu'un entier n se termine par X si et seulement si le reste de sa division par 100 est justement X.

Contrainte :  $10 \leq X \leq 99$

#### Exercice 6 :

Écrivez la fonction qui détermine si un nombre nb est une puissance de N.

Exemples : 27 est une puissance de 3 mais 15 n'est pas une puissance de 3 (il ne faut pas confondre A est un multiple de B et A est une puissance de B).

**Exercice 7 :**

Écrivez la fonction qui détermine si nb est le carré d'un nombre. La fonction retourne le nombre dont il est le carré, et -1 sinon.

**Exercice 8 :**

Un entier positif u est dit un rep-unit si son écriture en base 10 n'est formée que de chiffres valant 1. Par exemple, 111 est un rep-unit mais 42 n'est pas un rep-unit. Le nom rep-unit provient de la contraction des mots anglais REPeat et UNIT.

Écrire une fonction récursive estRepunit(n) qui teste si un entier positif n est un rep-unit.

Par exemple, estRepunit(0) et estRepunit(42) doivent renvoyer 0 et estRepunit(111) doit renvoyer 1  
Contrainte :  $n > 0$

**Exercice 9 :**

Écrivez une fonction récursive nbChiffres (n) qui renvoie le nombre de chiffres (décimaux) de l'entier n. Par exemple, nbChiffres(2030) vaut 4.

**Exercice 10 :**

Observez le motif ci-dessous formé de 10 entiers. On part d'un entier  $a \geq 0$ .

Dans l'exemple suivant,  $a = 81$ , et on construit une suite d'entiers ainsi :

81 80 40 20 10 5 4 2 1 0

Écrivez la procédure récursive qui permet d'afficher une telle suite de nombres.

## Exercices sur les tableaux à 1 dimension

**Exercice 11 :**

Écrivez la fonction qui détermine la valeur maximale contenue dans un tableau d'entiers.

**Exercice 12 :**

Écrivez la procédure qui inverse les éléments d'un tableau d'entier.

**Exercice 13 :**

Écrivez la fonction qui détermine si un tableau d'entier est trié de manière croissante.

**Exercice 14 :**

Écrivez la fonction qui détermine si tableau de nombre d'entiers est un palindrome.

Exemples :

0, 1, 2, 3, 4, 5, 6, 7, 8, 9 n'est pas un palindrome

0, 1, 2, 3, 4, 4, 3, 2, 1, 0 est un palindrome

**Exercice 15 :**

Écrivez la fonction qui détermine le nombre d'occurrences d'une valeur dans un tableau d'entiers.

**Exercice 16 :**

Écrire une fonction récursive qui renvoie vrai si les éléments d'un tableau d'entiers se suivent en alternant les signes (un nombre positif suivi d'un nombre négatif suivi d'un nombre positif, etc.

Exemples : [-5, 2, -4, 7] retourne 1, [3, -5, 2, -4, 7] retourne 1, [-2] retourne vrai également, 1 [-5, 2, 4, -7] retourne 0.

**Exercice 17 :**

On souhaite effectuer un tirage du loto. Écrivez une procédure récursive tirageLoto (int res [], int nbNumeros) qui tire aléatoirement 6 nombres entiers distincts compris entre 1 et 49.

**Exercice 18 :**

Écrire une fonction récursive contient01 (int t[], int taille) qui indique si tous les éléments d'un tableau d'entiers contient uniquement des 0 ou des 1

Exemples : {0, 1, 0} -> 1

{1, 1, 2} -> 0

{0, 0} -> 1

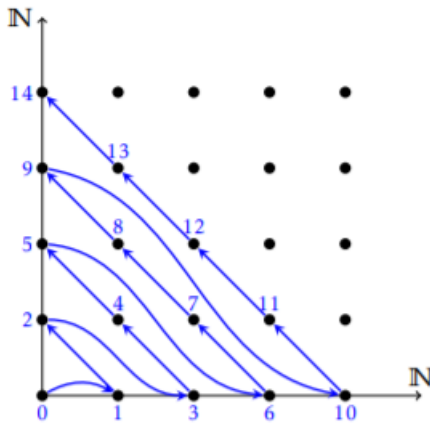
{0} -> 1

{1, 0, 0, 2, 0} -> 0

{ } -> 1

**Pour aller plus loin...Exercice 20 : (Exercice non obligatoire)**

Le couplage de Cantor consiste à numéroté la grille des éléments de  $\mathbb{N} \times \mathbb{N}$  en 0, 1, 2, 3, ..., comme le montre le dessin ci-dessous :



Le couplage de Cantor consiste à numéroté la grille des éléments de  $\mathbb{N} \times \mathbb{N}$  en 0, 1, 2, 3, ..., comme le montre le dessin ci-dessous :

Par exemple, le couple (2, 6) en colonne d'indice 2 et ligne d'indice 6 est numéroté 42. La numérotation commence en (0, 0) et s'effectue suivant des diagonales orientées sud-est vers nord-ouest. Arrivée sur la colonne la plus à gauche, la numérotation se poursuit sur la diagonale montante suivante en commençant par la ligne inférieure de la grille.

Écrire une fonction récursive `couplageCantor (x, y)` qui part des coordonnées d'un point et qui retourne le numéro qui lui est associé.

Exemple, `couplageCantor(2, 6) = 42`.