

TP N° 1 (La récursivité)

Exercice 1: L'exponentiation rapide

Écrire une fonction récursive d'entête **def puissance(x,n)** qui retourne la valeur de x^n en utilisant la formule suivante:

$$\text{puissance}(x, n) = \begin{cases} x, & \text{si } n = 1 \\ \text{puissance}(x^2, n/2), & \text{si } n \text{ est pair} \\ x \times \text{puissance}(x^2, (n-1)/2), & \text{si } n > 2 \text{ est impair} \end{cases}$$

Exercice 2: Conversion binaire <---->décimal

Écrire une fonction récursive d'entête **def Base2(n)** qui permet de convertir un entier naturel n dans le système binaire. La suite binaire est stockée dans une liste.

Exemple :	
<code>>>> base2 (6)</code> <code>[1 , 1 , 0]</code>	<code>>>> base2 (15)</code> <code>[1 , 1 , 1,1]</code>

Écrire une fonction récursive d'entête **def Base10(a)** qui retourne la valeur décimale d'une suite binaire passée comme argument (on traite deux cas a est un entier et a est une liste) .

Exemple :	
<code>>>> base10 ([1,1,0])</code> 6	<code>>>> base10 (1111)</code> 15

Exercice 3: Opérations sur les listes

1. Ecrire une fonction récursive de nom **TailleListe** qui prend en argument une liste et retourne le nombre d'éléments qu'elle contient.
2. Ecrire une fonction de nom **AffListe** qui affiche les éléments de la liste passée en paramètre.
3. Ecrire une fonction récursive de nom **AffListeInv** qui affiche, dans l'ordre inverse, les éléments de la liste passée en paramètre.
4. Ecrire une fonction **InsOrd** qui retourne une liste triée obtenue par l'insertion d'un nouvel élément à une liste triée.

Remarque:

On s'interdira d'utiliser les méthodes prédéfinies sur liste: append, sort, len, reverse,

Exercice 5: Opérations sur les chaînes de caractères

1. Ecrire une fonction récursive **strcpy(src,dest)** qui permet de copier une chaîne source dans une autre chaîne destination caractère par caractère.
2. Ecrire une fonction récursive **strcpm(ch1,ch2)** qui permet de comparer deux chaînes de caractères. Cette fonction retourne 1 si $ch1 > ch2$, retourne 0 si $ch1 = ch2$ et retourne -1 si $ch1 < ch2$.
3. Ecrire une fonction récursive **anagramme(ch1,ch2)** qui retourne **True** si une chaîne **ch1** donnée est anagramme d'une autre chaîne **ch2** ou **False** sinon. Par

exemple : 'algorithme' est anagramme de 'logarithme' et 'tanger' est anagramme de 'argent'

Exercice 4: Recherche dichotomique

On rappelle que l'idée de la recherche par dichotomie d'un élément x dans un tableau trié pour l'ordre croissant est la suivante :

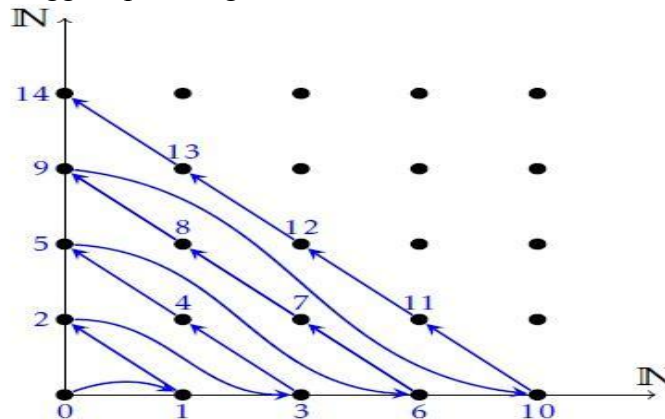
On compare x avec l'élément m du milieu du tableau,

- si $x = m$: on a trouve une solution ;
- si $x < m$: x ne peut pas se trouver après m dans le tableau ; il suffit alors de le rechercher dans la partie à gauche de m ;
- si $x > m$: x ne peut pas se trouver avant m dans le tableau ; il suffit alors de le rechercher dans la partie à droite de m ;

- a) Donner une fonction récursive **dico_recusif** qui traduit cet algorithme ?
b) Donner une fonction itérative **dico_iteratif** qui traduit cet algorithme?

Exercice 6: Récursivité croisée

L'ensemble $\mathbb{N} \times \mathbb{N}$ est dénombrable en numérotant chaque couple $(x, y) \in \mathbb{N}^2$ suivant le procédé suggéré par la figure ci-dessous :



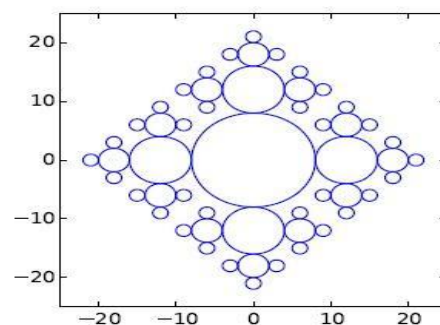
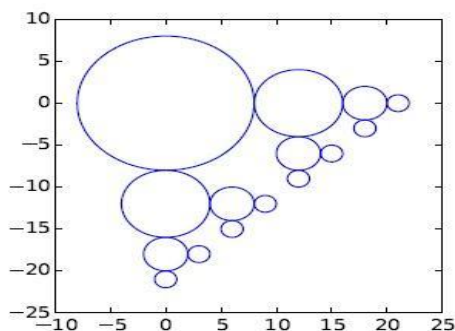
Ecrire une fonction récursive qui retourne le numéro du point de coordonnées (x, y).

Ecrire une fonction récursive qui retourne coordonnées (x, y) correspondant un numéro de point donné.

Exercise 7:

On suppose disposer d'une fonction `circle([x, y], r)` qui trace à l'écran un cercle de centre

(x, y) de rayon r. Définir deux fonctions récursives permettant de tracer les dessins présentés figure 1 (chaque cercle est de rayon moitié moindre qu'à la génération



précédente).

Le résultat des fonctions bubble1(4) et de bubble2(4)