

Réversivité

TP

Exercice 1: PGCD

- Écrire la fonction **pgcd(A,B)**, qui retourne le PGDC de deux entiers A et B en utilisant la méthode euclidienne :
 - Mettre dans R le reste de la division de A par B
 - Remplacer A par B et B par R.
 - Répéter jusqu'à ce que le reste soit nul
 - Le PGDC est le dernier reste non nul.
- Exemple :**
>>>pgcd(324,228)
12
- Écrire une fonction **réursive pgcd_rec(a,b)**, basée sur l'algorithme d'Euclide, qui calcule le pgcd de deux entiers naturels a et b.

Exercice 2: fonction mystere

Que fait la fonction suivante :

```
def mystere(n):  
    if (n > 1) : mystere (n//2)  
    print(n%2,end=' ')  
}
```

Quel est le résultat de l'appel : **mystere (13)**

Exercice 3: Recherche de chiffre

Définir une fonction qui, étant donné un entier positif x, et un chiffre n renvoie 1 si le chiffre n'apparaît dans x et 0 sinon.

1. En utilisant l'itération.
2. En utilisant la récursivité.

Exemple:

```
>>> present(85423, 7)  
False
```

```
>>> present(12345, 5)  
True
```

Exercice 4: Compter les chiffres

Écrire la fonction récursive **longueur(n)** qui retourne le nombre de chiffres d'un entier n.

Exemple:

```
>>> longueur (85423)  
5
```

Exercice 5: Conversion binaire <---->décimal

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

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

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 :	
>>> base10 ([1,1,0]) 6	>>> base10 (1111) 15

Exercice 6: 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 7: Opérations sur les chaînes de caractères

1. Ecrire une fonction récursive **strcpy(ch)** qui retourne une copie de la chaîne de caractères ch caractère caractère par caractère.
2. Ecrire une fonction récursive **inverse(ch)** qui retourne la chaîne inverse de ch.
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** : 'algorithm' est anagramme de 'logarithme' et 'tanger' est anagramme de 'argent'

Exercice 8: 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 trouvé 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;

b) Donner une fonction itérative **dico_iteratif** qui traduit cet algorithme?

a) Donner une fonction récursive **dico_recusif** qui traduit cet algorithme ?