```
from IPython.display import HTML
HTML(url="https://deptinfo-ensip.univ-poitiers.fr/FILES/NB/files/nb2.css")
```

1 Récursivité - exercices

Voici des éléments à avoir en tête :

- Prévoir les «cas de base», c'est à dire ceux qui ne nécessitent pas d'appel récursif de la fonction. S'il n'y a pas de cas de base, ça ne peut pas fonctionner.
- S'assurer (ça peut être très difficile) que les appels récursifs permettent, par la modification des paramètres, de se rapprocher des cas de base.
- Construire la valeur de retour en fonction des valeurs de retour des différents appels récursifs.

1.1 Exercices facile

1.1.1 Somme à compléter

Replacez la ligne 3 (pass) par ce qu'il faut pour que la fonction renvoie la somme de ses arguments (qui doivent être 2 nombres entiers positifs).

```
def somme(a, b):
    if b==0:
        pass
    return somme(a + 1 ,b - 1)
```

1.1.2 Produit de 2 entiers

Sur le même principe que l'exercice précédent, proposez un algorithme récursif qui calcule le produit de 2 entiers.

1.1.3 Puissance

Écrivez une fonction récursive qui élève un nombre r à la puissance n, avec n entier positif.

1.1.4 Plus grand diviseur commun

Écrivez une fonction de calcul récursive du calcul du Pgdc (algorithme d'Euclide)

1.1.5 Logarithme

Écrivez une fonction récursive qui calcule le logarithme discret d'un nombre. Le logarithme discret d'un nombre n en base b est le nombre m qui vérifie :

$b^m \leq n < b^{m+1}$

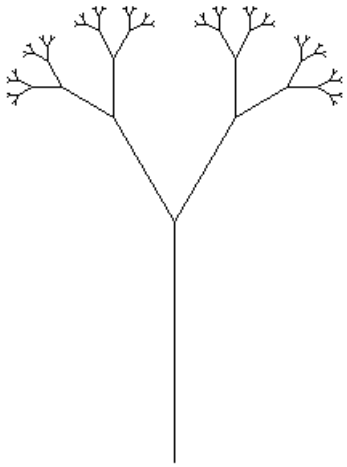
1.1.6 Palindromes

Écrivez une fonction récursive qui teste si une chaîne de caractères données est un palindrôme.

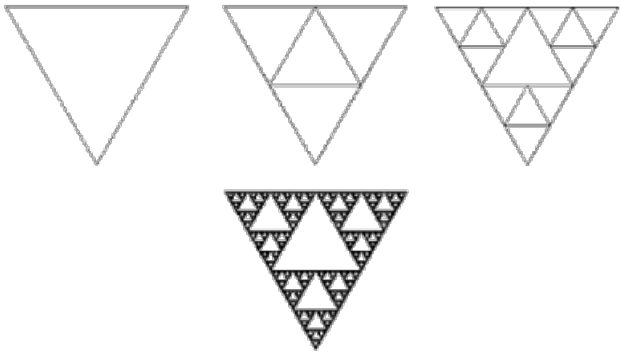
1.2 Module Turle

1.2.1 Courbe de Koch

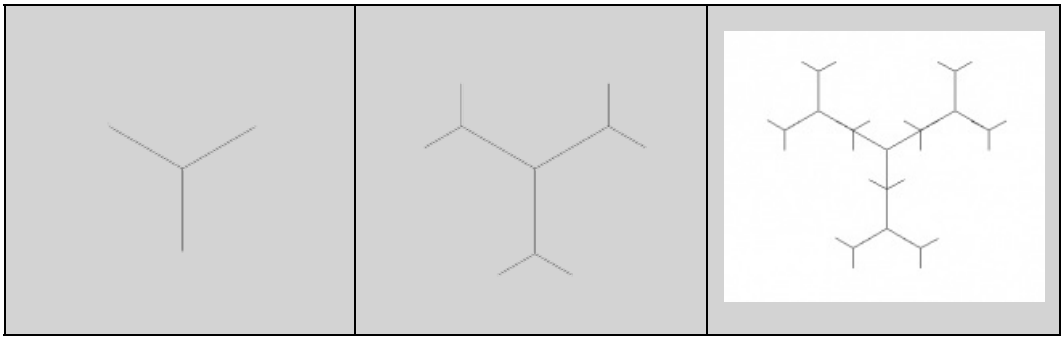
1.2.2 Arbre simple

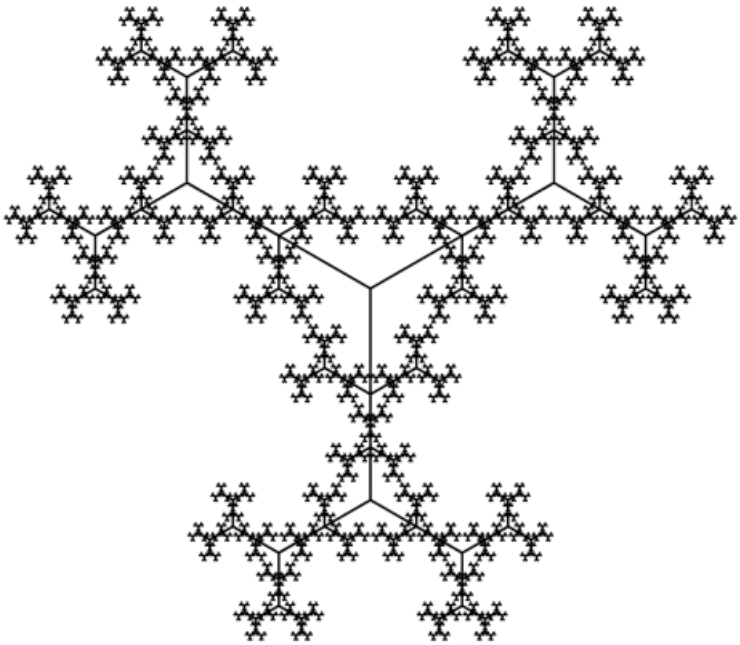


1.2.3 Triangle de Sierpinsky



1.2.4 Courbe étoile





1.3 Plus difficile

1.3.1 Recherche de chemin

Soit une liste t de n cases contenant uniquement des 0 et des 1. La première et la dernière case contiennent nécessairement des 0. Voici un exemple :

```
[0,0,1,0,0,0,1,1,0,1,1,1,0,0,0]
```

Écrivez une fonction qui trouve un chemin joignant la première à la dernière case, en n'empruntant que des cases contenant un 0, et en effectuant uniquement des déplacements de 1 ou 5 cases vers la gauche ou vers la droite.

Pour indiquer le chemin trouvé, la fonction devra renvoyer la liste des cases à emprunter dans l'ordre. Votre procédure devra naturellement fonctionner pour n'importe quelle séquence (pour laquelle il existe une solution). S'il n'y a pas de solution, votre fonction devra renvoyer une liste vide.

Exemples:

```
chemin([0, 0, 1, 0, 0, 0, 1, 1, 0, 1, 1, 1, 0, 0, 0]) => [0, 5, 4, 3, 8, 13, 14]
```

1.3.2 Par ici la monnaie

Par ici la monnaie

1.3.3 Le lion de Némée

Le lion de Némée

1.3.4 Des choix Cornéliens

Des choix Cornéliens

1.3.5 Séquence Génomique

Séquence Génomique

Ce pb devrait être traité par programmation dynamique. Demandez des infos sur le sujet si vous voulez le faire

1.4 Encore plus difficile...

1.4.1 Hydre de Lerne

Hercule et l'Hydre du Lerne

1.4.2 Le vaisseau du collectionneur

Le vaisseau du collectionneur