

## Chapitre 0 : Révision Python

## Série N°1 – Exercices extraits de concours

**Problème 1 :**

Un nombre premier est un entier qui n'est divisible que par 1 et par lui-même. On se propose d'écrire un programme qui établit la liste de tous les nombres premiers compris entre 1 et 100, en utilisant **la méthode du crible d'Eratosthène** :

- Créer une liste de 100 éléments, chacun initialisé à la valeur 1,
- Parcourir cette liste à partir de l'élément d'indice 2: si l'élément analysé possède la valeur une, mettez à zéro tous les autres éléments de la liste, dont les indices sont les multiples de l'indice auquel vous êtes arrivé.

Lorsque vous aurez parcouru ainsi toute la liste, les indices des éléments qui seront restés à 1 seront les nombres premiers recherchés.

En effet :

A partir de l'indice 2, vous annulez tous les éléments d'indices pairs 4, 6, 8...

Avec l'indice 3, vous annulez les éléments d'indices 6, 9...

Et ainsi de suite...

Seul resteront à 1 les éléments dont les indices sont effectivement des nombres premiers.

1. Ecrire une fonction **init(L)** permettant d'initialiser une liste de 100 éléments à un.
2. Ecrire une fonction **multiple(L, i)** permettant de mettre à zéro les éléments dont l'indice est un multiple d'un entier donné comme paramètre.
3. Ecrire une fonction **suivant(L, i)** qui à partir d'un indice donné, retourne le premier indice dont l'élément correspondant est différent de zéro.
4. Ecrire une fonction **crible(L)** permettant d'appliquer la méthode présentée par la crible d'Eratosthène pour retourner une liste qui contient les nombres premiers (il suffit d'étirer jusqu'à la racine entière de 100).
5. Ecrire une fonction récursive **crible\_recursive** permettant d'appliquer la méthode présentée pour retourner la liste initiale après les modifications.
6. Donner le schéma d'exécution de la fonction précédente (si on veut déterminer les entiers premiers qui sont inférieur à 10).
7. Ecrire un programme qui fait appel aux fonctions déjà définies afin d'afficher les entiers premiers compris entre 1 et 100.

**Problème 2 :**

On définit une suite de polynôme SP par :

$$\begin{cases} SP_0(x) = P(x) \\ SP_1(x) = -P'(x) \\ SP_{k+2}(x) = Q(x) * SP_{k+1}(x) + SP_k(x) \end{cases}$$

Avec :  $P(x)$  un polynôme de degré  $n$  ( $n \neq 0$ ) tel que :  $P(x) = \sum_{i=0}^n a_i x^i$

$P'(x)$  le polynôme dérivé de degré  $n-1$  tel que :  $P'(x) = \sum_{i=0}^{n-1} (i+1) a_{i+1} x^i$

$Q(x)$  est le produit des deux polynômes  $SP_{k+1}(x)$  et  $SP_k(x)$

Tout polynôme de degré «  $n$  » sera représenté sous forme d'une liste de taille «  $n+1$  » tel que les coefficients sont les éléments et les degrés sont les indices

**Exemple :**

La liste correspondante au polynôme  $P(x) = 1 + x - 5x^2 + x^3 - 2x^4 + 6x^6$  est :  $L = [1, 1, -5, 1, -2, 0, 6]$

On désire déterminer à partir de la suite ci-dessus le polynôme  $SP_m(x)$  avec  $m \geq 2$ .

1. Ecrire une fonction appelée **saisie\_deg( )** qui permet de saisir un entier strictement positif.
2. Ecrire une fonction appelée **saisie\_poly(n)** permettant de saisir la représentation d'un polynôme  $P(x)$  de degré «  $n$  » dans une liste.
3. Ecrire une fonction appelée **derive(P)** permettant de déterminer la représentation du polynôme dérivé d'un polynôme, représenté par une liste, le résultat sera une liste.
4. Ecrire une fonction appelée **opp\_poly(P)** permettant de déterminer la représentation du polynôme opposé, représenté par une liste, le résultat sera une liste.
5. Ecrire une fonction appelée **add\_poly(P1, P2)** qui prend comme paramètres deux polynômes  $P1$  et  $P2$  et retourne la représentation du polynôme  $P1 + P2$ .
6. Ecrire une fonction appelée **mul\_poly(P1, P2)** qui prend comme paramètres deux polynômes  $P1$  et  $P2$  ainsi que leurs degrés et retourne la représentation du polynôme  $P1 * P2$   
 Sachant que si  $P1(x) = \sum_{i=0}^{n1} a_i x^i$  et  $P2(x) = \sum_{i=0}^{n2} b_i x^i$   
 Alors le produit est  $P1(x) * P2(x) = \sum_{k=0}^{n1+n2} c_k x^k$  avec  $c_k = \sum_{i+j=k} a_i b_j$
7. Ecrire le programme principal permettant de :
  - a. Saisir le degré «  $n$  » d'un polynôme.
  - b. Saisir la liste qui représente le polynôme  $P$ .
  - c. Déterminer et afficher la liste  $D$  relative à la dérivée du polynôme  $P$ .
  - d. Déterminer et afficher la liste  $O$  relative à l'opposé du polynôme  $P$ .
  - e. Déterminer et afficher la liste  $A$  relative à l'addition de  $P$  et sa dérivée.
  - f. Déterminer et afficher la liste  $M$  relative à la multiplication de  $P$  avec sa dérivée.