TD 5 - Polynômes

Un polynôme est un objet de la forme

$$P = \sum_{k=0}^{n} a_k X^k = a_0 + a_1 X + a_2 X^2 + \cdots + a_n X^n,$$

où $n \in \mathbb{N}$ et $\forall i \in [\![0, n]\!], a_i \in \mathbb{R}$.
Si $a_n \neq 0$, n est le degré de P et on note $n = \deg P$.

Soient $P = \sum\limits_{k=0}^{n} a_k X^k$ et $Q = \sum\limits_{k=0}^{m} b_k X^k$. Quitte à rajouter des a_k nuls pour $k > n$ ou des b_k nuls pour $k > m$, on peut écrire $P = \sum\limits_{k=0}^{N} a_k X^k$ et $Q = \sum\limits_{k=0}^{N} b_k X^k$ où $N = \max(n, m)$.
On peut alors définir la somme

$$P + Q = \sum_{k=0}^{N} (a_k + b_k) X^k.$$

On peut également définir le produit

$$P \times Q = \sum_{k=0}^{2N} c_k X^k,$$

avec

$$\forall k \in [\![0, 2N]\!], c_k = \sum_{i+j=k} a_i b_j.$$

Exemple d'application de cette dernière formule : si $N = 3$, on a :

$(a_0 + a_1 X + a_2 X^2 + a_3 X^3)(b_0 + b_1 X + b_2 X^2 + b_3 X^3) =$
$a_0 b_0 + (a_0 b_1 + a_1 b_0) X + (a_0 b_2 + a_1 b_1 + a_2 b_0) X^2 + (a_0 b_3 + a_1 b_2 + a_2 b_1 + a_3 b_0) X^3 + (a_1 b_3 + a_2 b_2 + a_1 b_3) X^4 +$
$(a_2 b_3 + a_3 b_2) X^5 + a_3 b_3 X^6$

En Python, on va représenter un polynôme $P = \sum\limits_{k=0}^{n} a_k X^k$ par la liste $[a_0, a_1, \ldots, a_n]$.

1 – Ecrire une fonction `polynomenul` qui à une liste représentant un polynôme renvoie `True` si le polynôme est nul, `False` sinon.

2 – Ecrire une fonction `degre` qui à une liste représentant un polynôme P renvoie le degré $\deg P$.

3 – Ecrire une fonction `derivee` qui à une liste représentant un polynôme P renvoie la liste représentant la liste représentant le polynôme P' dérivée du polynôme P.

4 – Ecrire une fonction `somme` qui à deux listes représentant des polynômes P et Q renvoie la liste représentant le polynôme $P + Q$.

5 – Ecrire une fonction `produit` qui à deux listes représentant des polynômes P et Q renvoie la liste représentant le polynôme $P \times Q$.

Polynômes creux

On appelle *polynôme creux* un polynôme dont le rapport nombre de coefficients non nuls / degré est "petit", comme $X^{1000} - X^3$,. On préfère représenter de tels polynômes par une liste de couples, chaque couple correspondant à un monôme, et constitué de la valeur du coefficient et de la puissance correspondante. Ainsi, le polynôme ci-dessus sera représenté par `[(-1,3),(1,1000)]`. On considère que chaque puissance ne peut apparaître qu'une fois.

Dans un premier temps, on considère que les puissances ne sont pas triées.

1 – Ecrire une fonction `degre_c` qui à un polynôme creux non nul renvoie le degré du polynôme.

2 – Ecrire une fonction `coefficient_c` telle que si `p` est un polynôme creux et `i` est un entier, `coefficient_c(p,i)` renvoie le coefficient de degré `i` (éventuellement nul) du polynôme `p`.

3 – Ecrire une fonction `somme_c` qui renvoie la somme de deux polynômes creux.

4 – Ecrire une fonction `produit_elem_c` qui renvoie le produit d'un monôme par un polynôme creux.

5 – Ecrire une fonction `produit_c` qui renvoie le produit de deux polynômes creux, et qui utilise `produit_elem`.

6 – Ecrire une fonction `nettoie_nul` qui à un polynôme creux avec des coefficients éventuellement nuls (issus par exemple des calculs précédents) renvoie le polynôme creux sans coefficient nul correspondant.

7 – Ecrire une fonction `trie_c` qui à une liste avec des couples représentant un polynôme creux renvoie la liste avec les puissances dans l'ordre croissant.

8 – Ecrire une fonction `nettoie_c` qui à une liste avec des couples représentant un polynôme creux, présentant éventuellement plusieurs couples avec la même puissance, renvoie la liste mais sans puissance apparaissant deux fois (on considère qu'on somme les coefficients en cas d'occurrence multiple).

9 – A présent, on considère que les puissances sont dans l'ordre croissant. Réécrire les fonctions précédentes (sauf `trie_c`), en commençant par les plus simples à écrire.

10 – Discuter de la complexité des fonctions ci-dessus.