TP 3 (listes) : Arithmétique des polynômes

I Introduction

On représente un polynôme par la liste de ses coefficients, rangés dans l'ordre croissant des degrés. Par exemple, le polynôme $X^3 + 2X - 1$ est représenté par la liste `[-1; 2; 0; 1]` . Le polynôme nul est représenté par la liste vide.

Une telle liste est dite **propre** quand son dernier terme est non nul ou quand elle est vide. Un polynôme a une et une seule représentation par une liste propre, mais a une infinité de représentations impropres : la liste `[1;0;2;3;0;0]` représente par exemple aussi le polynôme $1 + 2X^2 + 3X^3$.

Dans tout ce sujet, on ne considèrera que des polynômes à coefficients **entiers**.

Toutes vos fonctions doivent être récursives.

Un petit conseil : vous pouvez bien sûr à chaque instant utiliser dans la définition d'une fonction une fonction écrite précédemment...

Afin de limiter la création d'usines à gaz particulièrement nocives l'énoncé précise les fonctions qui doivent s'écrire **sans** fonction auxiliaire.

II Opérations usuelles

1. Écrire une fonction `eval p x` , de paramètres une liste `p` représentant un polynôme p et un entier `x` , qui calcule l'évaluation $p(x)$ en x du polynôme p. **Pas de fonction auxiliaire**.
   Tester votre fonction sur des cas triviaux, puis un peu plus complexes.
2. Écrire une fonction `prod_ext z p` , de paramètres `z` un entier et `p` une liste représentant un polynôme p, qui calcule la liste associée au produit $z.p$ (i.e. la multiplication du polynôme p par le scalaire z). **Pas de fonction auxiliaire**.
   Tester votre fonction sur des cas triviaux, puis un peu plus complexes.
3. Écrire une fonction `somme p q` qui calcule une liste représentant le polynôme $p + q$. **Pas de fonction auxiliaire**.
   Tester votre fonction sur des cas triviaux, puis un peu plus complexes.
4. Écrire une fonction `produit p q` qui calcule une liste représentant le produit des polynômes $p \times q$. **Pas de fonction auxiliaire**.
   Tester votre fonction sur des cas triviaux, puis un peu plus complexes.
5. Écrire une fonction `puissance p k` qui calcule une liste représentant le polynôme p^k. **Pas de fonction auxiliaire**.
   Tester votre fonction sur des cas triviaux, puis un peu plus complexes.
6. Écrire une fonction `compose p q` qui calcule une liste représentant le polynôme $p \circ q = p(q)$. **Pas de fonction auxiliaire**.
   Tester votre fonction sur des cas triviaux, puis un peu plus complexes.
7. Les fonctions précédentes peuvent éventuellement construire des représentations impropres. Écrire une fonction `nettoyage p` qui construit la représentation propre d'un polynôme. Vous pourrez vous servir de la fonction prédéfinie `rev` qui construit la liste miroir d'une liste : `rev [1;5;4] = [4;5;1]` .
   Tester votre fonction sur des cas triviaux, puis un peu plus complexes.

III Division euclidienne

Soit a et b deux polynômes, b unitaire ($i.e$ son coefficient dominant est 1) : dans ce cas, on montre que le quotient et le reste de la division euclidienne de a par b sont aussi à coefficients entiers. On supposera donc toujours implicitement que b est unitaire.

1. Écrire une fonction `coeff_dom p` , qui calcule le coefficient dominant du polynôme p.
   Tester votre fonction sur des cas triviaux, puis un peu plus complexes.
2. On veut écrire un algorithme récursif qui calcule q, r, deux polynômes, tels que $a = bq + r$ et $\deg r < \deg b$. À quelle condition simple sur les degrés, q et r sont-ils déterminables sans calcul ?
3. On écrit $a = \alpha + Xc$ où α est un entier et c un polynôme et on suppose qu'on sait calculer le quotient et le reste q_1, r_1 de la division euclidienne de c par b.
   - 3-a Montrer que si $\deg r_1$ est assez petit, le reste r de la division euclidienne de a par b est $Xr_1 + \alpha$.
   - 3-b Dans le cas contraire, montrer qu'il existe un entier β tel que $r = Xr_1 - \beta b + \alpha$.
4. Écrire une fonction `division_euclidienne a b` , qui calcule le couple (q, r). **Pas de fonction auxiliaire**.
   Tester votre fonction sur des cas triviaux, puis un peu plus complexes. Vous pouvez aller chercher sur internet des exemples et vérifier que tout va bien (et éventuellement trouver des erreurs dans ce qui est proposé sur internet...)

IV Pgcd

1. Écrire une fonction `pgcd a b` de paramètres a, b deux polynômes, qui calcule **un** pgcd de a et b.
   On rappelle que les calculs doivent être faits avec des entiers et uniquement avec des entiers !. On pourra tirer profit de la remarque suivant : dans l'arithmétique des polynômes, on travaille à une constante multiplicative près...
2. Améliorer l'algorithme pour qu'il donne en plus des solutions à l'équation de Bezout
   $$au + bv = d$$
   où d est un pgcd de a et b.