TP D'OPTION INFORMATIQUE 2
Manipulation de polynômes en Caml

Dans toute la suite, on représente le polynôme réel $a_n X^n + \ldots + a_1 X + a_0$ par la liste de flottants $[a_0; \ldots; a_n]$ (on notera bien que les coefficients sont listés selon les puissances croissantes de X).

1 Opérations élémentaires sur les polynômes

1. Écrire une fonction `produit_externe` prenant en argument un flottant et un polynôme et renvoyant leur multiplication.
2. Écrire une fonction `somme` calculant la somme de deux polynômes.
3. Lorsqu'on considère la division euclidienne d'un polynôme p par X, comment s'expriment le quotient et le reste, et à quoi correspondent-ils du point de vue de la représentation par liste ?
4. En déduire une fonction `evaluation` prenant en argument un polynôme p et un flottant v et renvoyant l'évaluation de p en v (ie la valeur $p(v)$).
5. Écrire une fonction `produit` calculant le produit de deux polynômes $p1$ et $p2$. On raisonnera à nouveau à partir d'une division euclidienne par X.
6. Écrire une fonction `afficher` prenant en argument un polynôme et affichant à l'écran son écriture mathématique. Par exemple, `afficher [1. ; 0. ; 3. ; -2.];;` pourrait afficher à l'écran `X^0 + 3.X^2 - 2.X^3`.

   On utilisera une fonction récursive auxiliaire prenant en argument la liste des coefficients restant à afficher, ainsi que la puissance de X associée au premier élément de cette liste.
7. Écrire une fonction `derivation` calculant le polynôme dérivé d'un polynôme donné.

2 Normalisation

On dit que la représentation d'un polynôme est normale si son dernier terme, s'il existe, n'est pas nul. Ainsi, $X^2 - 2$ a pour représentation normale `[-2.; 0.; 1.]`, mais peut aussi être représenté par `[-2.; 0.; 1.; 0.; 0.]`. Le polynôme nul a pour représentation normale `[]`.

1. Écrire une fonction `est_normal` testant si une liste est une représentation normale.
2. Écrire une fonction `normalisation` prenant un polynôme en argument et renvoyant sa représentation normale.

3 Interpolation de Lagrange

Écrire une fonction `lagrange` prenant en argument deux **tableaux** de flottants `x` et `y` de même longueur, et calculant le polynôme d'interpolation de Lagrange valant `y.(i)` en `x.(i)` pour chaque indice i. On rappelle que ce polynôme est donné par la formule suivante :

$$P = \sum_{j=0}^{n} y_j \left(\prod_{\substack{i=0 \\ i \neq j}}^{n} \frac{X - x_i}{x_j - x_i} \right)$$

On renverra ce polynôme sous forme normalisée.

4 Arithmétique des polynômes

Pour implémenter la division euclidienne de deux polynômes, il est beaucoup plus aisé de manipuler une représentation sous forme de tableau. Il nous faut donc tout d'abord des fonctions permettant de traduire une liste en tableau et réciproquement.

1. Écrire une fonction `array_of_list` traduisant une liste en tableau.
2. Écrire une fonction `list_of_array` traduisant un tableau en liste.
3. Écrire une fonction `div_eucl` prenant en argument deux polynômes, et renvoyant le quotient et le reste de la division euclidienne du premier par le second. Les polynômes donnés en arguments sont représentés par des listes, qui seront traduites en tableaux au sein de la fonction. Les quotient et reste calculé seront renvoyés sous forme de listes normalisées.
4. En déduire une fonction `pgcd` renvoyant un pgcd de deux polynômes.