

Algorithmes et Structures de Données 2

Projet polynôme

On désire pouvoir manipuler des polynômes de degré quelconque. Pour effectuer ce travail :

- chaque terme $a \cdot x^n$ du polynôme sera représenté par une entité contenant la puissance entière n et le coefficient réel a
- le polynôme sera représenté par une liste de terme triée dans l'ordre décroissant des puissances des termes.

Les listes seront implémentées par une structure chaînée de cellules avec les primitives vue en cours : `liste_vide`, `tete`, `corps`, `insere_en_tete`, `supprime_en_tete`, `affiche`, `insere_en_position`, `supprime_en_position`. Vous pourrez ajouter d'autres primitives en fonction de vos besoins.

Les primitives à implémenter pour les polynômes sont :

- **`def saisie_polynome()`** : renvoie un polynôme saisi au clavier sous forme d'une liste de terme triée par ordre décroissant des puissances des termes
- **`def affichage(poly)`** : affiche à l'écran le polynôme **`poly`**
- **`def valeur(poly, x)`** : renvoie la valeur de **`poly`** pour un **`x`** donné
- **`def addition(poly1, poly2)`** : renvoie un polynôme résultat de l'addition de **`poly1`** et **`poly2`**
- **`def soustraction(poly1, poly2)`** : renvoie un polynôme résultat de la soustraction de **`poly1`** et **`poly2`**
- **`def multiplication(poly1, poly2)`** : renvoie un polynôme résultat de la multiplication de **`poly1`** et **`poly2`**
- **`def division(poly1, poly2)`** : renvoie un polynôme résultat de la division de **`poly1`** par **`poly2`** ([ici](#) et [là](#), [Wikipédia](#))

Pour ces fonctions, il est conseillé de faire plusieurs versions : récursive avec les primitives des listes, itératives avec les primitives des listes, itératives en accédant directement à la structure de données des listes (cellules et pointeur suivant). Lors de l'examen, il y aura obligatoirement un exercice pour chacune de ces versions.

Le projet est à réaliser en Python 3.