

## Projet C : Polynômes

*Avant de commencer : la qualité des commentaires, avec notamment la présence des antécédents, des conséquents, des invariants de boucle, les rôles de chacune des fonctions, ainsi que les noms donnés aux variables, l'emploi à bon escient des majuscules et la bonne indentation rentreront pour une part importante dans l'appréciation du travail. Ce projet doit permettre de montrer votre autonomie et votre compréhension tant dans la conception du programme que dans sa réalisation. Enfin, si les codes de plusieurs projets se trouvent être identiques, ou être copiés depuis le web, tous les projets concernés seront immédiatement sanctionnés par un zéro.*

### 1 Travail à réaliser

L'objectif de ce projet est de réaliser un ensemble de fonctions permettant la manipulation de polynômes. Vous développerez un programme en langage C permettant l'utilisation de ces fonctions par interaction avec un utilisateur pour permettre :

1. la saisie d'un ou plusieurs polynômes ;
2. la saisie de l'opération à effectuer ;
3. l'affichage du résultat d'une opération.

### 2 Les polynômes

Dans ce projet, nous considérerons des polynômes, combinaison linéaire des produits de puissances à une inconnue. Un polynôme  $p$  de degré  $n$  (positif) à une inconnue  $x$  aura la forme suivante :

$$p(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_i x^i + \dots + a_1 x + a_0$$

**Note :** par la suite, nous considérerons que les coefficients  $a_i$  sont des entiers.

Les polynômes à lire sur l'entrée standard (un par ligne) seront une suite d'addition ou de soustraction de monômes. Un monôme suit la syntaxe  $[[-]/[a]][x[^\sim d]]$ , où  $a$  et  $d$  sont des entiers positifs. Les crochets indiquent un terme optionnel. Ainsi les monômes suivants correspondent à cette syntaxe :

7, -890, -4x<sup>3</sup>, 2x, 345x<sup>23</sup>

Sur l'entrée standard, les monômes peuvent apparaître dans **n'importe quel ordre de degré**, avec éventuellement plusieurs monômes de même degré. Des espaces séparateurs peuvent également apparaître dans l'écriture du polynôme. Ainsi, les entrées suivantes correspondent à des polynômes valides :

```
5x^3 - 2x - 7
-9x^4+3x^2-x
-3x^2 + 23x + 2x^2 - 3x^4 + 1
2x + 4x^3 - x + -3x^5
3+-x
5-6
```

### 3 Fonctions sur les polynômes

Vous programmerez un ensemble de fonctions qui permettront la manipulation des polynômes. Une liste non exhaustive (d'autres opérations pourront être définies si nécessaire) est donnée ci-dessous :

- Initialisation ;
- Lecture et écriture ;
- Affectation du polynôme (*i.e.* lui donner un nom)
- Évaluation de  $p(x)$  par la méthode de Horner pour un  $x$  donné ;
- Addition ;
- Soustraction ;
- Multiplication ;
- Dérivation ;
- Division euclidienne ;
- Factorisation par  $(x - \alpha)$ ,  $\alpha$  un entier donné.

Les fonctions d'initialisation permettront l'initialisation à 0 des coefficients d'un polynôme, et l'initialisation d'un polynôme  $p1$  à partir d'un polynôme  $p2$  (fonction de copie).

La fonction *lire* lit sur l'entrée standard un polynôme qui suit la syntaxe donnée précédemment. La fonction *écrire* écrit sur la sortie standard un polynôme en partant de son degré le plus élevé. Ainsi, pour les polynômes donnés dans l'exemple précédent, la fonction d'écriture produira :

```
5x^3 - 2x - 7
-9x^4 + 3x^2 - x
-3x^4 - x^2 + 23x + 1
-3x^5 + 4x^3 + x
-x + 3
-1
```

Les fonctions *additionner*, *soustraire*, *multiplier* et *diviser*, implémentent respectivement l'addition, la soustraction, la multiplication, et la division entre deux polynômes.

La fonction *dérivée* retourne le polynôme dérivé du polynôme passé en paramètre.

La fonction *factoriser* factorise un polynôme. Dire qu'un polynôme  $p(x)$  est factorisable signifie qu'il existe deux polynômes  $q(x)$  et  $r(x)$  de degré supérieur ou égale à 1 tels que  $p(x) = q(x) \times r(x)$ .

## 4 Structure de données

Toutes les fonctions précédentes manipulent des polynômes et vous serez amenés à définir un type C `polynome` pour les représenter. Les coefficients d'un polynôme `p` pourront être mémorisés dans un tableau d'entiers de telle façon que `p[i]` est le coefficient du degré `i`. Vous pourrez définir le type `polynome` par la déclaration :

```
#define DEGREMAX 30 /* par exemple */
typedef int polynome[DEGREMAX];
```

Une variable `p` de type `polynome` sera simplement déclarée comme suit :

```
polynome p;
```

## 5 Bonus

Vous pourrez ajouter les fonctions suivantes à votre programme :

- plus grand commun diviseur de deux polynômes ;
- plus petit commun multiple de deux polynômes.

## 6 Exemple d'exécution

Votre programme devra proposer à l'utilisateur les opérations à exécuter sur les polynômes lus. Voici un simple exemple d'exécution :

```
cmd> p := 2x^3+3x^2
cmd> q := x^2+x^3
cmd> p + q
p + q = 3x^3+4x^2
cmd> p := p + q
p = 3x^3+4x^2
cmd> p'
p' = 9x^2+8x
cmd> p := x^5+3x^3+2x^2+2x+2
cmd> q := x^2+1
cmd> p / q
p / q = (x^3+2x+2, 0)
cmd> r := 2x^8-6x^7+4x^3-12x^2
cmd> fact(r,3)
(x-3.00)(2x^7+4x^2)
```

Ressources :

- [perso.univ-rennes1.fr/michel.coste/B05\\_Polynomes.pdf](http://perso.univ-rennes1.fr/michel.coste/B05_Polynomes.pdf)
- [fr.wikiversity.org/wiki/Polyn%C3%B4me/Arithm%C3%A9tique\\_des\\_polyn%C3%B4mes](http://fr.wikiversity.org/wiki/Polyn%C3%B4me/Arithm%C3%A9tique_des_polyn%C3%B4mes)
- [allken-bernard.org/pierre/mpsi/cours/Ch18.S6.xhtml](http://allken-bernard.org/pierre/mpsi/cours/Ch18.S6.xhtml)

## 7 Remise du projet

Votre projet est à faire en binôme. Il est à rendre au plus tard :

**mercredi 18 janvier 2017 à 12h – aucun délai ne sera accordé –**

sous forme d'une archive `polynome-n1-n2.tar.gz` avec `n1` et `n2` sont les deux noms des étudiants qui forment le binôme, et que vous enverrez à vos enseignants aux adresses suivantes : `vg@unice.fr` et `luc.hohwiller@unice.fr`.

### 7.1 Cette archive devra contenir :

- le fichier source (`.c`) correctement documenté (chaque fonction doit avoir un commentaire, les invariants de boucle doivent être marqués), indenté, et codé (les noms de variables explicites, éviter les trop longues fonctions) ;
- un fichier `Documentation` au format pdf et décrivant le fonctionnement général du programme, les algorithmes, ainsi que les choix de programmation ;
- la compilation avec les options `-Wall -pedantic` ne doit pas donner de *warning*.