

## Programmation 2

### Feuille de TP n°6 (version révisée)

#### Pointeurs, structures et tableaux (dynamiques)

section\*Exercice 2. Structures et tableaux dynamiques Cet exercice reprend en le complétant l'exercice 2 de la feuille de TP n°6 posté sur Moodle Dans cet exercice, vous implémenterez la structure `polynome` et les fonctions de l'exercice 1 de la feuille de TD n°5.

Écrivez un programme qui

1. lit deux entiers  $d_p, d_q \geq -1$  passés en arguments du programme sur la ligne de commande ;
2. initialise les coefficients de deux polynômes  $p$  et  $q$ , de degré respectivement majoré par  $d_p$  et  $d_q$  ;
3. affiche le polynôme  $p$  puis vérifie l'égalité de Bargmann-Fock sur  $p$ , c'est-à-dire
$$d(X \cdot p(X))/dX - X \cdot dp(X)/dX = p(X) ;$$
4. affiche le polynôme  $q$  puis calcule et affiche le produit des polynômes  $p$  et  $q$  ;
5. lit un flottant  $a$  entré au clavier par l'utilisateur puis calcule et affiche la valeur de  $p(a)$  ;
6. calcule et affiche le quotient et le reste de la division euclidienne du polynôme  $p$  par le polynôme  $q$  (dont le coefficient du monôme de plus haut degré aura été normalisé à 1) ; **[N.B. Ce calcul ne doit être abordé que si tous les autres points de la feuille de TP ont été traités avec succès.]**
7. libère avant la fin de son exécution l'ensemble de l'espace mémoire qu'il a alloué sur le tas.

#### Consignes pour l'écriture du programme.

- Organisez votre code en trois fichiers : un fichier d'entêtes `polynome.h` contenant la définition de la structure `polynome` et les déclarations des fonctions des diverses fonctions sur les polynômes, un fichier de fonctions `polynome.c` contenant les définitions des fonctions déclarées dans `polynome.h` et un fichier « principal » contenant la fonction principale. Vous trouverez sur Moodle un exemple de fichier `polynome.h` et le fichier `Makefile` permettant de construire un exécutable à partir des trois fichiers susmentionnés. [Cf. les diapos du cours n°5 (première partie, présentée en amphi le 2 avril dernier)].
- Pour l'affichage des polynômes (étapes 3 et 4), vous devez écrire une fonction qui reçoit l'adresse d'un polynôme en paramètre d'entrée et affiche le polynôme en s'approchant du modèle ci-dessous, pour l'affichage du polynôme  $8 - X + 7X^3$ 
$$7X^3 - X + 8$$
- Pour l'initialisation pseudo-aléatoire des polynômes (étape 2), vous devez écrire une fonction qui reçoit un entier  $d$  positif ou nul en paramètre d'entrée et qui initialise les coefficients d'un polynôme de degré  $d$  à des valeurs entières pseudo-aléatoires comprises entre  $-9$  et  $+9$  après avoir alloué l'espace nécessaire sur le tas en appelant la fonction `creer_polynome` de la question 1.a de l'exercice 1 de la feuille de TD n°5, et renvoie l'adresse du polynôme ainsi « créé » et initialisé.
- Pour la multiplication par un scalaire, la dérivation, l'addition et la multiplication des polynômes (étapes 3 et 4), le programme doit appeler les fonctions des questions 2.a à 2.d de l'exercice 1 de la feuille de TD n°5.
- La consigne du TD s'applique : l'espace mémoire alloué par les fonctions que vous écrivez est limité à ce qui est nécessaire pour le stockage des coefficients.

- Pour l'évaluation du polynôme  $q$  en  $a$ , c'est-à-dire le calcul de  $q(a)$ , vous devez écrire une fonction, que votre programme appellera, qui évite de calculer (de façon redondante)  $a^2$  puis  $a^3$  puis  $a^4$ , etc. Pour cela, il suffit de remarquer que

$$a_0 + a_1 x + a_2 x^2 + \cdots + a_{n-1} x^{d-1} + a_n x^d = a_0 + x(a_1 + x(a_2 + \cdots + x(a_{n-1} + x a_n) \cdots))$$

- Pour la libération de l'espace alloué sur le tas (étape 5), le programme doit appeler la fonction `destruire_polynome` de la question 1.c de l'exercice 1 de la feuille de TD n°5.