Programmation 2

Feuille de TD-TP n°6

(Structures, tableaux, pointeurs et allocation dynamique sur le tas)

Exercice 1 (TD). Opérations sur les polynômes à coefficients entiers

On cherche à implémenter quelques opérations sur les polynômes univariés à coefficients entiers. Un polynôme de $\mathbb{Z}[X]$

 $p(X)=a_0+a_1\,X+a_2\,X^2+\cdots+a_i\,X^i\cdots$ (les coefficients a_i tous nuls à partir d'un certain rang) sera représenté par une structure :

```
struct polynome {
    int * coefficients;
    int taille;
    int degre;
};
```

dont le champ coefficients contient l'adresse d'un bloc d'entiers stockant les coefficients de p (le coefficient a_i en position i dans le bloc); le champ taille contient la taille du bloc alloué pour les coefficients de p et le champ degre contient le degré de p. Par convention, le degré d'un polynôme nul est -1. Dans la suite, le mot « polynôme » renverra toujours à une structure telle que définie ci-dessus et l'expression « taille du polynôme » désignera la taille du bloc alloué pour ses coefficients.

Question 1. Écrivez la définition de chacune des trois fonctions décrites ci-après.

- a) creer_polynome reçoit un entier n en paramètre d'entrée, alloue l'espace mémoire pour un polynôme nul de taille n, et renvoie son adresse.
- b) reformater_polynome reçoit l'adresse d'un polynôme en paramètre d'entrée et réalloue pour ses coefficients un bloc de taille exactement adaptée à son degré, si cette réallocation est nécessaire; cette fonction ne renvoie pas de valeur et ne doit pas faire appel à la fonction realloc.
- c) detruire_polynome reçoit un pointeur vers l'adresse d'un polynôme p en paramètre d'entrée et libère tout l'espace mémoire alloué sur le tas pour ce polynôme; cette fonction ne renvoie pas de valeur.

Question 2. Écrivez la définition de chacune des quatre fonctions décrites ci-après **en respectant la consigne suivante** : l'espace mémoire alloué est limité à ce qui est nécessaire pour le stockage des coefficients.

- a) scalaire_polynome reçoit l'adresse d'un polynôme p et un entier a en paramètres d'entrée. Cette fonction ne renvoie pas de valeur et à la fin de son exécution, tous les coefficients du polynôme p ont été multipliés par a.
- b) additionner_polynome reçoit les adresses de deux polynômes p et q en paramètres d'entrée et renvoie l'adresse du polynôme p+q après lui avoir alloué l'espace mémoire adapté.
- c) multiplier_polynome reçoit les adresses de deux polynômes p et q en paramètres d'entrée et renvoie l'adresse du polynôme $p\cdot q$ après lui avoir alloué l'espace mémoire adapté.
- d) deriver_polynome reçoit l'adresse d'un polynôme p en paramètre d'entrée et renvoie l'adresse du polynôme dérivé d p(X)/dX après lui avoir alloué l'espace mémoire adapté.

Exercice 2 (TP). Manipulation de polynômes à coefficients entiers

Écrivez un programme qui

- 1. lit deux entiers d_{max} et d_{min} ($d_{max} \ge d_{min} \ge -1$) passés en arguments du programme sur la ligne de commande;
- 2. initialise les coefficients de deux polynômes p et q, respectivement de degrés d_{max} et d_{min} ;
- 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. 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.

• 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 <code>creer_polynome</code> de la question 1.(a) de l'exercice 1, 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.
- 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 la libération de l'espace alloué sur le tas (étape 5), le programme doit appeler la fonction detruire_polynome de la question 1.(c) de l'exercice 1.