

Les boucles sont interdites.

**Exercice 1. Polynômes (faisable en TP)**

On souhaite représenter les polynômes en stockant la liste des coefficients et des exposants sous forme d'une liste chaînée. Par exemple, le polynôme  $2x^3 - 5x^7$  sera représenté par la liste  $((2,3), (-5,7))$ . On remarque donc que, par rapport aux listes d'entiers, chaque maillon contient ici deux entiers. La liste `null` représentera le polynôme nul (au sens 0). On remarque au passage que la représentation d'un polynôme n'est pas unique. Par exemple, le polynôme nul est représenté par la liste `null`, et aussi par la liste  $((0,3))$  (correspondant à  $0x^3$ ).

On obtient donc la classe suivante.

```
class Polynome{
    private int coeff;
    private int deg; //deg >= 0
    private Polynome suiv;

    public Polynome(int c, int d, Polynome p){
        coeff = c;
        deg = d;
        suiv = p;
    }
}
```

Si vous êtes en TP, pensez à rajouter "public static" dans l'en tête de toutes les méthodes.

**Question 1.1.**

Ecrire une fonction `String toString(Polynome p)` qui retourne une chaîne représentant  $p$  sous le format suivant : si  $p = ((2,3), (-5,7))$ , alors on doit retourner `"+2x^3-5x^7"`. Bonus : améliorer cette version en enlevant le premier "+" en trop.

**Question 1.2.**

Ecrire une fonction `int eval(Polynome p, int x)` qui calcule la valeur de  $p$  en  $x$ . Par exemple, si  $p = 2x^3 - 5x^7$ , alors `eval(p,2)` doit retourner  $2 * 2^3 - 5 * 2^7 = -624$ .

**Question 1.3.**

Ecrire une fonction `int degre(Polynome p)` qui calcule le degré du polynôme (c'est à dire le plus gros exposant). La méthode a pour prérequis que le polynôme est réduit (c'est à dire que tous les exposants apparaissant sont différents). En effet dans le cas contraire, si l'on avait par exemple  $p = 2x^3 + x - 2x^3$ , il serait difficile de détecter qu'il faut retourner 1, et pas 3. Attention aux coefficients nuls : si  $p = 4x^2 + 0x^5 + 2x^3$ , il faut retourner 3, et pas 5.

**Question 1.4.**

Ecrire une fonction `Polynome simplifie(Polynome p)` qui calcule un nouveau polynôme indépendant en enlevant tous les monômes ayant un coefficient de 0. Par exemple, avec  $p = 2x^3 + 0x^4 - 10x - 2x^3$  on doit retourner  $2x^3 - 10x - 2x^3$ .

**Question 1.5.**

Ecrire une fonction `Polynome reduire(Polynome p)` qui calcule un nouveau polynôme en fusionnant les monômes de même degrés. Par exemple, avec  $p = 1 + 3x^2 + x + x^2$  on doit retourner  $1 + 4x^2 + x$  (l'ordre des monômes n'est pas imposé). Vous pouvez vous écrire une fonction auxiliaire, que vous devrez spécifier.