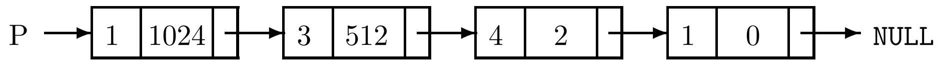
Opérations sur les polynômes

Le but de cet exercice est d'implémenter les opérations élémentaires sur les polynômes à coefficients entiers. Tout polynôme sera représenté par une liste de monômes. Un monôme correspond donc à une structure de type `struct monome` comportant trois champs : son coefficient de type `int`, son degré de type `int` et un pointeur sur le monôme suivant. Le type `polynome` utilisé dans toute la suite est donc un pointeur sur la structure `struct monome`.

Par exemple, le polynôme $P(x) = 1 + 4x^2 + 3x^{512} + x^{1024}$ correspond à la liste chaînée :



1 Définition du type polynôme

Définir le type polynôme dans un fichier `type_poly.h` qui sera inclus dans les autres fichiers source utilisant le type `polynome`.

2 Construction d'un polynôme

Afin de construire un polynôme, écrire une fonction qui insère un monôme en tête d'un polynôme. Son prototype sera :

```
polynome constructeur(int c, int d, polynome P)
```

Cette fonction, tout comme la fonction d'affichage, sera définie dans un fichier contenant le corps des fonctions élémentaires sur les listes, `listes.c`.

3 Affichage d'un polynôme à l'écran

Écrire une fonction qui prend comme paramètre un polynôme et qui l'affiche à l'écran sous la forme :

```
1+5x+8x^7-9x^12+x^23
```

4 Vérification des fonctions précédentes

On vérifiera les fonctions de construction et d'affichage en utilisant la fonction `lecture_polynome` qui prend comme paramètre un nom de fichier (de type `char *`) et qui retourne un objet de type `polynome` correspondant au polynôme décrit dans le fichier. Le nom du fichier donnant le polynôme pourra être passé en argument de la fonction `main`. Le contenu de ce fichier sera une suite de caractères sans espace du type :

```
x^23+1+5x+8x^7-9x^12
```

La fonction `lecture_polynome` est définie dans le fichier `lecture_polynome.c` disponible sur le site Web du cours

```
http://www-rocq.inria.fr/codes/Anne.Canteaut/COURS_C/
```

On y trouvera également le fichier en-tête `lecture_polynome.h` correspondant.

Écrire un programme principal de test et un `Makefile` permettant de produire l'exécutable.

Les fonctions suivantes pourront être regroupées dans un même fichier source.

5 Degré d'un polynôme

Écrire une fonction qui prend comme paramètre un polynôme et qui retourne son degré. Cette fonction aura pour prototype :

```
int degre(polynome P)
```

Dans les questions suivantes, on supposera que les polynômes manipulés ont des monômes de degrés distincts qui sont classés dans l'ordre croissant des degrés.

6 Addition de deux polynômes

Écrire une fonction qui prend comme paramètres deux polynômes triés et qui en retourne un troisième égal à leur somme :

```
polynome add_poly(polynome P1, polynome P2)
```

Le polynôme retourné sera également un polynôme trié dont tous les monômes sont de degrés distincts.

7 Produit de deux polynômes

Écrire une fonction qui retourne le produit de deux polynômes. On pourra utiliser la fonction `add_poly` ainsi qu'une fonction intermédiaire effectuant le produit d'un polynôme par un monôme. Le prototype de cette dernière fonction sera :

```
polynome produit_monome(int x, int deg, polynome P)
```