Feuille de TD - II

Exercice 1 (polynomes, version 1).
On manipule des polynômes P[x] à coefficients et variables réels. Les coefficients du polynôme sont rangés dans un vecteur par puissances croissantes : l'élément d'indice 0 stocke le terme constant du polynôme, celui d'indice 1 le terme de degré 1, etc, jusqu'au degré du polynôme. Si le polynôme est de degré `n`, le vecteur des coefficients doit donc avoir une taille `n+1`. Par exemple, le polynôme $1 - 3x^2 + 5x^4$ est représenté par le vecteur :

1	0	-3	0	5

c'est-à-dire $1 + 0x^1 - 3x^2 + 0x^3 + 5x^4$

Le polynôme nul (de degré 0 et coefficient égal à 0) est représenté par le vecteur `{ 0.0 }`.

On définit le type polynôme de la façon suivante :
```
typedef vector<double> Polynome ;
```

(1) Écrire une fonction qui renvoie le degré d'un polynôme : `size_t degre(Polynome p)`.

(2) Écrire une fonction qui prend en entrée un polynôme `p` et une valeur `v` et renvoie la valeur de `p` évalué en `v`. L'en-tête est : `double valeur(Polynome p, double v)`.
Exemple : Le polynôme $1 - 3x^2 + 5x^4$ évalué en 2 vaut $1 - 3 * 2^2 + 5 * 2^4 = 69$.

(3) Écrire une fonction `Polynome somme(Polynome p1, Polynome p2)` qui renvoie la somme de `p1` et `p2` (somme terme à terme des deux polynômes). Attention : les deux polynômes n'ont pas forcément le même degré et certains termes peuvent s'annuler dans la somme. En particulier, la somme d'un polynôme et de son opposé doit renvoyer la représentation du polynôme nul. *Exemple* : la somme des polynômes $1 + 3x^2 + 5x^4$ et $-1 + 3x^1 - 2x^2 - 5x^4$ doit renvoyer la représentation de $3x^1 + 1x^2$, donc un vecteur de taille 3.
Note : la fonction `pop_back()` supprime le dernier élément d'un vecteur.

(4) Même question pour le produit de deux polynômes. On rappelle que le produit de deux termes ax^n et bx^m est le terme $a \times bx^{n+m}$ et que faire le produit de deux polynômes revient à multiplier tous les termes du premier polynôme par tous les termes du second. Par exemple, le produit de $3 + 2x^2 + 5x^3$ par $1 + 2x^1 + 3x^4$ vaut $3 + 6x^1 + 2x^2 + 9x^3 + 19x^4 + 6x^6 + 15x^7$. Bien sûr, dans le résultat il ne doit y avoir qu'un terme avec une puissance donnée, les coefficients doivent être rangés par puissances croissantes et le dernier coefficient du vecteur doit être non nul (sauf si le résultat est le polynôme nul).
En-tête de la fonction : `Polynome produit(Polynome p1, Polynome p2)`.

Exercice 2 (polynomes, version 2).
La représentation précédente est peu efficace puisqu'elle oblige à stocker des termes dont le coefficient est nul. Pour stocker le polynôme x^7 avec la représentation précédente on est obligé d'avoir un vecteur de taille 8 alors qu'il n'y a qu'un terme à représenter. Pour palier ce défaut, on définit un type qui représente les monômes de coefficient (non nul) avec leur exposant et un polynôme est un vecteur de tels monômes, classés par degrés croissants. La taille du vecteur n'est plus liée au degré du polynôme, et les exposants de deux éléments consécutifs du vecteur ne diffèrent plus forcément de 1. On définit les deux types ci-dessous :

```
typedef struct {
 double coef ;          // coef est maintenant forcément non nul
 size_t degre ;         // le degré du terme
} Monome ;              // exemple : 3x^5

typedef vector<Monome> Polynome ;
```

(1) Réécrire la fonction qui renvoie le degré d'un polynôme : `size_t degre(Polynome p)`.

(2) Réécrire la fonction qui évalue la valeur d'un polynôme en un point.
En-tête : `double valeur(Polynome p, double v)`.

(3) Réécrire la fonction pour la somme de deux polynômes.
En-tête : `Polynome somme(Polynome p1, Polynome p2)`.
Indication : on pourra s'inspirer de l'algorithme utilisé pour la fusion de tableaux triés.

(4) (*facultatif*) Réécrire la fonction pour le produit de deux polynômes :
On pourra écrire deux versions : l'une qui calcule le produit de tous les termes deux à deux et en fait la somme, la seconde qui calcule directement le coefficient de l'éventuel terme de degré `n` dans le résultat : ce terme est la somme des produits des coefficients des termes de degrés respectifs `i` et `j` dans `p1` et `p2` tels que `i+j = n`.
En-tête : `Polynome produit(Polynome p1, Polynome p2)`.