Univérsité Hassan II Casablanca

Ecole normale supérieure (ENS) Casablanca

Module : Structures de données en C (Info 4)

Année universitaire 2020/2021

Parcours 2éme année LEESM **Prof. Mohamed Lahby**

TP 5

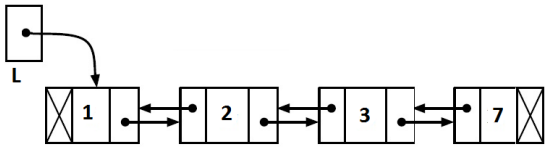
Liste doublement chaînée

Problème 1

On considère un polynôme à coefficients réels définit sous la forme suivante :

$$P(x) = a_0x^0 + a_1x^1 + a_2x^2 + ... + a_ix^i + ... + a_{n-1}x^{n-1} + a_nx^n$$

Dans ce problème, on chosit de représenter les polynômes sous la forme d'une liste doublement chaînée. Chaque cellule d'une liste contiendra le coefficient a_i, qui est un nombre réel.
Exemple : Pour $P(x) = 7X^3 + 3X^2 + 2X + 1$ on va créer une liste doublement chaînée contenant les 4 coefficients (degré+1) comme suit :



Un polynôme nul est représenté avec une liste chaînée vide.
On suppose avoir déjà crée dans la mémoire dynamique une liste doublement chaînée représentant un polynôme. Cette liste est définie comme suit :

```
typedef struct cellule{
    float coeff;
    struct cellule *preced;
    struct cellule *suiv;
}Polynome;
```

Question 1. Ecrire une fonction **int EstPolynomeNul(Polynome *P)** qui retourne 1 si P est un polynôme nul ou 0 sinon.

Question 2. Ecrire une fonction **Polynome * LitPolynome(Polynome *P)** qui permet de construire une liste chaînée contenant un polynôme (demande à l'utilisateur de taper le degré et les coefficients).

Question 3. Ecrire une procédure **void AffichePolynome(Polynome *P)** qui permet d'afficher un polynôme.
Exemple : Pour $P(x) = 2x^2 - 3x + 5$ on obtient l'affichage suivant :
$P(x) = 2x**2 - 3x**1 + 5**0$

Question 4. Ecrire une fonction **int DegrePolynome(Polynome* P)** qui retourne le degré du polynôme P. La fonction retournera -1 dans le cas du polynôme nul.

Question 5. Ecrivez la fonction **float EvalPoly(Polynome* P,float a)** ; qui retourne la valeur que vaut P au point a (c-à-d la valeur de P(a)).

Question 6. Ecrire une fonction **Polynome * SommePoly(Polynome* P1,Polynome* P2)** ; qui retourne un polynôme contenant la somme des deux polyômes.

Problème 2

Un polynôme est creux s'il possède très peu de coefficients non nuls par rapport à son degré (on peut imaginer un polynôme de degré 10000 ayant seulement 13 coefficients non nuls).
Exemple :

$$P(X) = 1 + 3X + X^{100}$$

Pour représenter un polynôme creux on peut utiliser une liste doublement chaînée de monômes. Chaque monôme possède trois champs (une strcuture contenant le coefficient et le degré, le champ suivant qui pointe sur un autre monôme et le champ precedant qui pointe sur un autre monôme).
Pour ce faire, nous définissons les deux structures suivantes :

```
typedef struct monome_elem{
        float coef;
        int degre;
}monome;
```

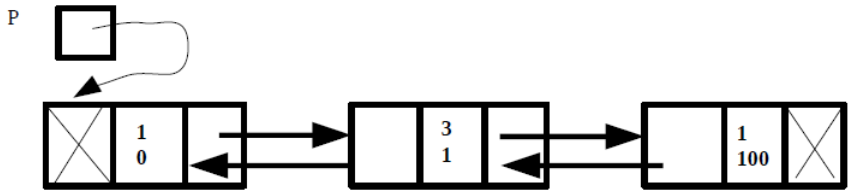
Un polynôme sera codé avec une liste doublement chaînée. Le type polynome est le suivant :

```
typedef struct elem{
        monome pval;
        struct elem * suivant;
        struct elem * precedent;
}polynome;
```

On dit que la liste représentant un polynome est en forme réduite si et seulement si :

- ✓ les monômes apparaissent dans la liste par ordre croisssant de degré
- ✓ la liste ne contient que des monômes avec des coefficients différents de 0

Exemple : Pour $P(x) = 1 + 3X + X^{100}$ on va créer la liste doublement chaînée suivante :



Dans toutes les questions qui suivent, on suppose que tous les polynômes passés en paramètres sont sous la forme de liste réduite. Les polynômes retournées par ces fonctions doivent aussi être sous la forme de liste réduite.

Question 1. Ecrire une procédure **void AffichePoly(polynome *P)** qui permet d'afficher un polynôme.
Exemple : Pour $P(x) = 5 - 3x + 2x^2$ on obtient l'affichage suivant :
$P(x) = 5 + 3x**1 + 2x**2$

Question 2. Ecrire une fonction **int DegrePoly(polynome* P)** qui retourne le degré du polynôme P.

Question 3. Ecrire une fonction **polynome * add(polynome* P1,polynome* P2)** ; qui retourne un polynôme contenant la somme des deux polyômes.

Question 4. Ecrire la fonction **float EvalPoly(polynome* P,float a)** ; qui retourne la valeur que vaut P au point a (c-à-d la valeur de P(a)).

Question 5. Ecrire la fonction **polynome * tab_to_poly(int n,float tab[])** qui prend en argument un tableau tab de n éléments et construit un polynome tel que pour tout indice i, tab[i] est le coefficient du monôme x^i .

Question 6. Ecrire la fonction : **polynome * supprimer_ieme(polynome * p, int i)** qui supprime le monôme de degré i du polynôme p.

Question 7. Ecrire une fonction **int existe(polynome * p, int d)** qui retourne 1 si un degré d figure dans le polynôme p ou 0 sinon.

Question 8. Ecrire la fonction : **polynome * add_monome(polynome * p, monome m)** qui retourne le polynôme p en insérant le monôme m dans la bonne position.

Question 9. Ecrire la fonction : **polynome * derivee(polynome * p)** qui retourne une nouvelle liste contenant la dérivée de p.