

QROC

Module A31TTN

Structures de données

Aucun document n'est autorisé, les réponses sont à donner sur le présent document.

Exercice 1 – Récursivité

On considère les deux fonctions suivantes :

```
int X(int n)
{
    if (n == 0)
        return 1;
    else
        return n*X(n - 1);
}
```

```
int Y(int n)
{
    int ret = 1;
    int i = 1;
    if (n == 0)
        ret=1;
    else{
        for(i = 1; i <= n; i++)
            ret = ret*i;
    }
    return ret;
}
```

1) Donnez leurs résultats pour $n=4$ et $n=5$. Que pouvez-vous remarquer ? Expliquez, en particulier, la trace d'exécution de la fonction X ?

2) Que font alors les fonctions X et Y ci-dessus ? Donnez des noms significatifs à ces fonctions ? Sans faire de calcul donnez leurs complexités respectives.

Exercice 2 – Listes chaînées

Dans cet exercice nous souhaitons être capable de représenter un nombre quelconque de polynômes différents tant qu'il y a de la mémoire disponible. En général, nous voulons représenter le polynôme :

$$P(x) = a_m x^{e_m} + a_{m-1} x^{e_{m-1}} + \dots + a_1 x^{e_1}$$

Où les a_i sont des coefficients réels non nuls et les e_i sont des exposants entiers positifs tels que $e_m > e_{m-1} > \dots > e_2 > e_1 > = 0$.

Nous souhaitons représenter les polynômes sous forme de listes simplement chaînées. Une structure adéquate pour représenter un polynôme peut être définie de la façon suivante :

```
typedef struct noeud {  
    float coefficient;  
    float exposant;  
    struct noeud* lien;  
};  
typedef noeud *poly;
```

Où les champs coefficient et exposant désignent respectivement les coefficients et les exposants du polynôme.

Exemple : Soit le polynôme $P(x) = 3x^{14} + 2x^8 + 1$

Ce polynôme peut être représenté sous la forme suivante :



1) Ecrire une fonction ***poly integral(poly p)*** qui permet de calculer l'intégrale d'un polynôme.

Exemple :

Soit le polynôme $a(x) = 3x^{14} + 2x^8 + 1$

intergal(a) retourne $b(x) = 3/15x^{15} + 2/9x^9 + x$

Rappel : l'intégrale du monôme x^n est: $\frac{1}{n+1} x^{n+1}$.

2) Sachant que deux polynômes sont identiques si et seulement si ils ont les mêmes coefficients et les même exposants, écrire une fonction *int Egal(poly p, poly q)* qui permet de tester si deux polynômes sont identiques. *Egal* retourne 1 si les deux polynômes sont identiques, 0 sinon.

Exemple :

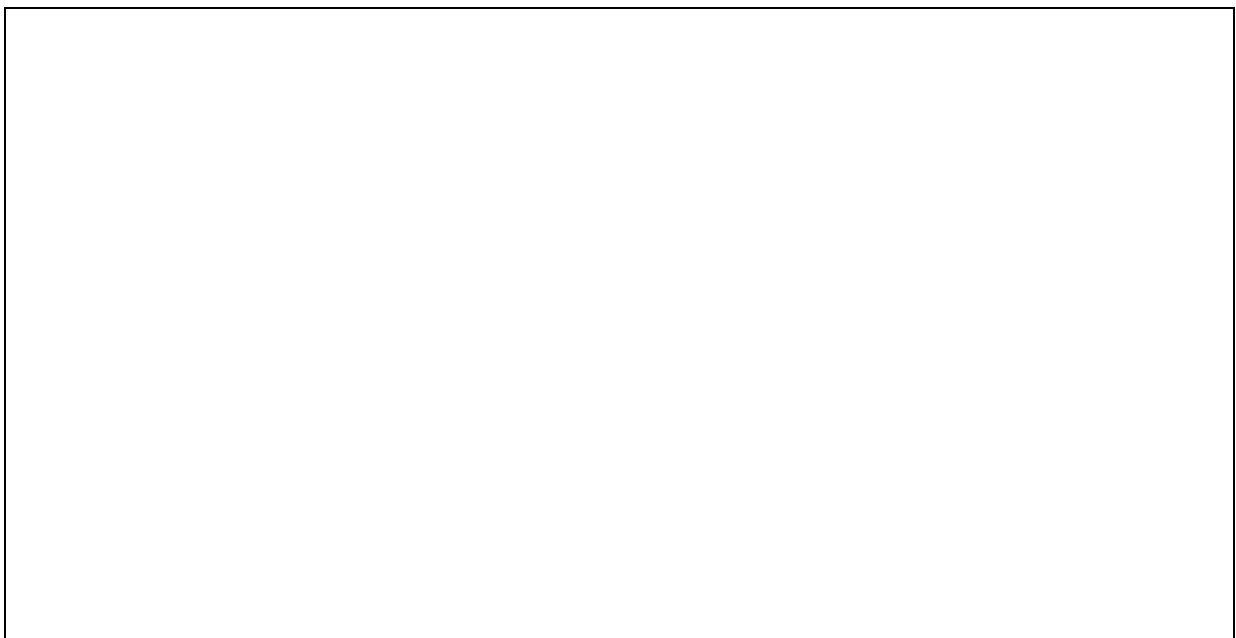
Soit le polynôme $a(x)=3x^{14}+2x^8+1$, soit $q(x)=3x^2+2x^8+1$, les polynômes p et q sont différents. *Egal* retourne 0.



3) Ecrire une fonction qui réalise la multiplication d'un polynôme par un scalaire *poly Multiplication(poly p, float x)*.

Exemple :

$$2 * a(x) = 6x^{14} + 4x^8 + 2$$



4) Ecrire une fonction qui réalise l'addition de deux monômes
poly AdditionMonome(poly p, poly q).

Exemples :

$$a(x) = 3x^4, b(x) = 2x^5, a(x)+b(x)=3x^4+2x^5$$

$$a(x) = 3x^{14}, b(x) = 2x^{14}, a(x)+b(x)=5x^{14}$$

5) Ecrire une fonction qui permet d'additionner deux polynômes
poly AdditionPolynome(poly p, poly q).

Exercice 3 – Les arbres binaires

On considère la suite de nombres suivante : 6 – 80 – 7 – 51 – 65 – 53 – 39 – 9.

- 1) Partant d'un arbre binaire vide, on ajoute successivement les valeurs de la suite ci-dessus. Donnez en justifiant votre raisonnement l'arbre binaire résultat (*il ne s'agit pas de donner la version équilibrée de l'arbre*).



2) Vérifier si l'arbre obtenu est équilibré, s'il n'est pas équilibré, partez d'un arbre vide et insérez la suite de nombre de telle façon que le résultat soit équilibré. Expliquez les opérations de rotation effectuées.



3) On considère la structure de données suivante pour représenter cet arbre :

```
typedef struct Arbre {  
    int Noeud;  
    struct Arbre * SAG;  
    struct Arbre * SAD;  
} Arbre;
```

3.1- Proposez une fonction récursive qui retourne la plus petite valeur de l'arbre.

3.2- Donnez une fonction qui permet de calculer le nombre des nœuds d'un arbre.

Bonne chance.