TP2: Les polynômes

Ibrahim ALAME

20/12/2023

Il s'agit de mémoriser des polynômes d'une variable réelle et de réaliser des opérations sur ces polynômes. Le nombre de monôme est variable, aussi une allocation dynamique s'impose. La gestion en liste facilite l'ajout ou la suppression de monômes pour un polynôme donné. Le type polynôme est défini de la façon suivante:

```c
#include <stdio.h>
#include <stdlib.h>
typedef struct polynome{
    int coeff;
    int exp;
    struct polynome *suivant;
} Polynome;
```

1. Commenter la fonction suivante:

```c
Polynome *addMonome(Polynome *p, int coefficient, int exposant){
    if(p==NULL || p->exp < exposant){
        Polynome * m = malloc(sizeof(Polynome));
        m->coeff=coefficient;
        m->exp=exposant;
        m->suivant=p;
        return m;
    }
    if (p->exp == exposant){
        p->coeff += coefficient;
        if(p->coeff==0 && p->suivant != NULL){
            Polynome *asupprimer=p;
            p=p->suivant;
            free(asupprimer);
        }
        return p;
    }else{
        p->suivant = addMonome(p->suivant,coefficient,exposant);
        return p;
    }
}
```

2. Écrire une fonction de destruction d'un polynôme donné p en supprimant bien les pointeurs des termes de la liste dans le bon ordre pour ne perdre aucune information.

```
void destruction(Polynome * p){
    while (p!=NULL){
         ...
         ...
         ...
    }
}
```

3. Écrire une fonction `somme` permettant de calculer la somme polynomiale des polynômes p et q:

```
Polynome* somme(Polynome* p,Polynome* q){
    Polynome* s = NULL;
    while (p!=NULL){
        ... ;
        ... ;
    }
    while (q!=NULL){
        ... ;
        ... ;
    }
    return s;
}
```

4. Commenter la fonction suivante:

```
void afficherPolynome(Polynome *monome){
    while (monome != NULL) {
        printf("%c", (monome->coeff >= 0) ? '+' : '-');
        if(abs(monome->coeff)!=1 || monome->exp == 0)
            printf("%d", abs(monome->coeff));
        if (monome->exp >= 1)
            printf("X");
        if (monome->exp > 1)
            printf("^%d", monome->exp);
        monome = monome->suivant;
    }
    printf("\n");

}
```

5. A cette étape le programme principal ressemble à ceci:

```
int main(){
    Polynome *p = NULL, *q = NULL;
    p=addMonome(p,-5,0);
    p=addMonome(p,2,1);
    p=addMonome(p,2,2);
    p=addMonome(p,1,3);
```

```
    printf("P=");afficherPolynome(p);

    q=addMonome(q,5,0);
    q=addMonome(q,3,1);
    q=addMonome(q,1,2);
    printf("Q=");afficherPolynome(q);

    Polynome *s = somme(p,q);
    printf("P+Q=");afficherPolynome(s);
    destruction(p);
    destruction(q);
    destruction(s);
    return 0;
}
```

```
P=+X^3+2X^2+2X-5
Q=+X^2+3X+5
P+Q=+X^3+3X^2+5X+0
```

6. Écrire une fonction `produit` permettant de calculer le produit des polynômes p et q:

```
Polynome* produit(Polynome* p,Polynome* q){
    Polynome* r=NULL;
    while (p!=NULL) {
        Polynome *m=q;
        while (m != NULL) {
            ... ;
            ... ;
        }
        ... ;
    }
    return r;
}
```

7. Écrire une fonction `Puissance` récursive permettant de calculer une puissance entière selon le principe suivant:

$$x^n = \begin{cases} \left(x^2\right)^{\frac{n}{2}} & \text{si } n \text{ est pair} \\ \left(x^2\right)^{\frac{n-1}{2}} \times x & \text{si } n \text{ est impair} \end{cases}$$

8. Écrire une fonction `int Valeur(Polynome* p, int x0)` qui calcule la valeur d'un polynôme p en un point $x = x_0$ donné. Exemple: calculer la valeur de $P(x) = 3x^5 + 2x^3 + 1$, en $x = 2$:

```
P(x)= 3 x^5 +2 x^3 +1
Valeur de x? 2
P(2)= 113
```

9. Écrire une fonction `Polynome* diff(Polynome* p)` qui renvoie le polynôme dérivé d'un polynôme donné.

10. Peut-on modifier la structure de la liste polynômiale définie au début du problème pour qu'on puisse calculer un polynôme primitif s'annulant en $x = 0$ à coefficients rationnels sans utiliser les types float et double?

11. Écrire une fonction `divEuclidienne` permettant d'effectuer la division euclidienne d'un polynôme A par un polynôme B. La fonction `divEuclidienne` affichera ou retournera deux polynômes Q (quotient) et R (reste).