

## Devoir surveillé

**Matière :** Algorithmique et structures de données II

**Enseignants :** Majdi Jribi et Khaoula Bezzina

**Filière :** MPI

**Nombre de pages :** 2

**Semestre:** 2

**Date:** 01 Avril 2021

**Durée:** 1h30mn

**Documents :** non autorisés

**Chaque exercice doit être rédigé sur une feuille d'examen à part.**

### Exercice 1 (4 pts)

Ecrire en **langage C** une fonction récursive **int chiffre(int n, int k)** qui permet de retourner le  $k^{\text{ème}}$  chiffre à partir de la droite d'un entier positif **n**.

**Exemple :**  $n=5678$ ,  $k=3$ , le résultat est 6

### Exercice 2 (16 pts)

On se propose de développer un ensemble de fonctions en **langage C** permettant de manipuler des polynômes creux. Un polynôme creux est un polynôme contenant très peu de monômes non nuls.

**Exemple :**  $P(x) = 5.6 x^{1280} + 0.8 x - 9$  contient **1281** termes dont **3** seulement sont non nuls.

Un monôme est défini par la structure suivante:

```
typedef struct sMonome {  
    int deg;  
    double coef;  
    struct sMonome *suiv;  
} TMonome;
```

Chaque monôme est décrit par une structure de données de type **TMonome** comportant les 3 champs suivants :

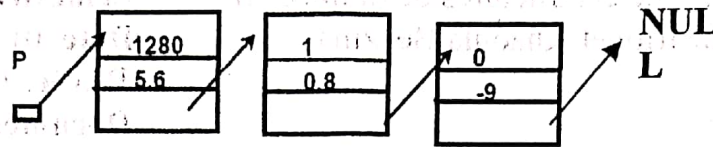
- **deg** : entier représentant le degré du monôme ;
- **coef** : réel représentant le coefficient du monôme ;
- **suiv** : pointeur sur le monôme suivant.

Un polynôme est défini par la structure de données suivante :

```
typedef struct sPolynome {  
    TMonome *prem;  
} TPolynome;
```

Un polynôme sera décrit par une structure **TPolynome** contenant un seul champ **prem** :

- Si **prem** est NULL, le polynôme correspondant sera le polynôme zéro ;
- Si **prem** est non NULL, il pointera sur le monôme de plus haut degré dont le coefficient est non nul. Les monômes de coefficient non nuls sont chaînés par ordre de degré décroissant.



Une illustration du polynôme P

1- Ecrire en langage C la fonction itérative :

**void Ajout\_Poly(TPolynome \*resultat, int deg, double coef)**

qui ajoute à un polynôme la valeur d'un monôme défini par son degré (deg) et son coefficient (coef).

2- Ecrire en langage C la fonction itérative :

**void Addition\_Poly(TPolynome \*resultat, TPolynome oper1, TPolynome oper2)**

qui permet d'additionner deux polynômes donnés oper1 et oper2.

3- Ecrire en langage C la fonction itérative :

**void Derivation\_Poly(TPolynome \*p, TPolynome polynome)**

qui permet de calculer la dérivée du polynôme donné polynome.

4- Ecrire en langage C la fonction itérative :

**void Primitive\_poly(TPolynome \*p, TPolynome polynome)**

qui permet de calculer une primitive du polynôme donné polynome.

5- Ecrire en langage C la fonction itérative :

**void Multiplication\_Poly(TPolynome \*resultat, TPolynome oper1, TPolynome oper2)**

qui permet de multiplier deux polynômes donnés oper1 et oper2.