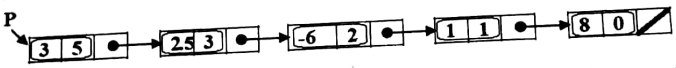
Université Ibn Tofail
Faculté des Sciences
Département d'Informatique
Kénitra

A.U. 2019/2020
Filière : SMI/SMA
Semestre 4
Structures de Données/ Informatique 4

Examen (1h)

Exercice 1:
On veut comparer les deux structures de données **Tableaux** et **listes chaînées**, décrire les avantages et les inconvénients de chacune des structures.

Exercice 2 :
On convient de représenter les polynômes par des listes linéaires chaînées triées dans l'ordre décroissant des puissances des monômes. Un monôme est représenté par un couple de valeur : un exposant (**exp**) qui est un entier et le coefficient correspondant (**coef**) un réel. Les monômes sont rangés dans la liste par ordre décroissant des exposants. Par exemple le polynôme $\mathbf{P(x) = 3\ x^5 + 2.5\ x^3 - 6\ x^2 + x + 8}$ est représenté par la liste P suivante :



Dans ce problème, on propose de définir une structure de données « **Polynome** » permettant la gestion de la liste des monômes **P**. Les opérations prévues sur cette structure de données sont :

- **creerPoly**() : permet de créer une structure vide.
- **degre**(P) : retourne le degré du polynôme **P** donné en paramètre.
- **evaluer**(P , x) : qui retourne la valeur du polynôme **P** pour un réel **x** donné.
- **rechMonome**(P, k) : permet de vérifier l'existence d'un monôme d'exposant **k** dans la liste. Elle retourne l'adresse du maillon contenant le monôme s'il existe sinon elle retourne **NULL**.
- **suppMonome**(P, k) : supprime le monôme d'exposant **k** s'il existe dans **P.**
- **ajoutMonome**(P, c, k) : permet d'alimenter la liste **P** par le monôme d'exposant **k** et coefficient **c** selon les cas suivants :
  - Si l'exposant **k** n'existe pas, le monôme sera ajouté.
  - S'il existe, l'opération additionne la valeur **c** au coefficient du monôme d'exposant **k**, un monôme de coefficient nul doit être supprimé. L'ajout sera fait par ordre décroissant des exposants.
- **somme**(P1 , P2): retourne la somme des deux polynômes **P1** et **P2**.
- **derive**(P) : retourne le polynôme égal à la dérivée du polynôme **P**.

1. Donner la spécification abstraite de la structure de données **Polynome**.

2. Implémenter les opérations précédentes en utilisant les deux structures de données **Monome** et **Polynome** définies par :

```
typedef struct monome{
      float coef ; int exp ;
      } Monome ;
typedef struct polynome{
      Monome info; struct monome *suiv ;
      } Polynome ;
```

Université Ibn Tofail
Faculté des Sciences
Département d'Informatique
Kénitra

A.U. 2019/2020
Filière : SMI/SMA
Semestre 4
Structures de Données/ Informatique 4

Epreuve de rattrapage

(1h)

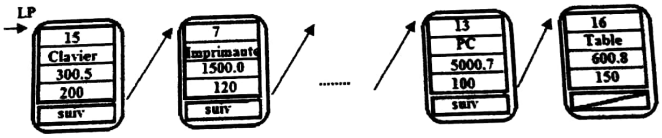
Exercice 1:

On veut définir un type abstrait permettant de manipuler les nombres complexes. Ce type sera appelé **Complexe** et devra comporter des opérations pour additionner et multiplier deux nombres complexes et pour calculer le module et le conjugué d'un nombre complexe.

1- Donner la spécification abstraite du type **Complexe** (avec axiomes et pré-conditions).
2- Définir une structure de données permettant de représenter le type abstrait **Complexe**.
3- Implémenter les opérations précédentes.

Exercice 2 :

Un ensemble des produits est représenté par une liste **LP**. Chaque produit est caractérisé par : u identificateur **id** (entier), un **nom** (Chaîne de 10 caractères), un prix unitaire **pu** (réel) et une quantité e stock **qs** (entier).



Dans ce problème, on se propose de définir une structure de données « **LProduit** » permettant la ges de la liste des produits **LP**. Les opérations prévues sur cette structure de données sont :

- **creerListeProd**() : permet de créer une structure vide.
- **rechercherProd**(LP, idp) : permet de vérifier l'existence d'un produit d'identificateur **idp** da liste **LP**. Elle retourne l'adresse du maillon contenant le produit s'il existe sinon elle ret NULL.
- **ajouterProd**(LP, idp, nomp, prix, qte) : permet d'alimenter la structure **LP** par **qte** unit produit **'nomp'** ayant le prix unitaire **'prix'** et d'identificateur **idp**. Si le produit existe déjà d structure, l'opération d'ajout augmente sa quantité par **qte**. L'ajout sera à la fin de la liste.
- **supprimerProd**(LP, idp) : supprime un produit, s'il existe, de la structure **LP** l'identificateur est passé en paramètre.
- **totalStock** (LP, idc) : Calcule le prix total des produits en stock.
- **trierProd**(LP) : permet de trier les produits par ordre alphabétique de leurs noms.

1. Donner la spécification abstraite de la structure de données **LProduit** (avec pré-conditions).
2. On veut représenter la structure **LProduit** de liste de produits sous forme de liste simp chaînée. Définir les différentes structures de données utiles.
3. Implémenter les opérations précédentes.

Université Ibn Tofail
Dept. de Mathématiques, Kénitra

Année 2019-2020

Rattrapage Analyse numérique

SMA-SMI Semestre 4

Exercice 1 : Soient les matrices

$$A_1 = \begin{pmatrix} 1 & 2 & -2 \\ 1 & 1 & 1 \\ 2 & 2 & 1 \end{pmatrix} \text{ et } A_2 = \begin{pmatrix} 2 & -1 & 1 \\ 2 & 2 & 2 \\ -1 & -1 & 2 \end{pmatrix}$$

On cherche à résoudre les problèmes suivants :

$$(\mathcal{P}_i) \begin{cases} \text{trouver } x \in \mathbb{R}^3 \text{ tel que} \\ A_i x = b_i \end{cases}, \ i = 1, 2 \text{ et } b_i \in \mathbb{R}^3$$

par les méthodes de Jacobi et/ou Gauss-Seidel.

1) Quelle(s) méthodes(s) utiliseriez-vous pour $(\mathcal{P}_1)$?

2) Quelle(s) méthodes(s) utiliseriez-vous pour $(\mathcal{P}_2)$?

Exercice 2 :

On considère une fonction réelle f définie sur un intervalle $[a, b]$, contractante (Lipschitzienne de rapport $k \in]0, 1[$. Soit $x_0 \in [a, b]$; On définit la suite

$$x_{n+1} = f(x_n)$$

(a) Montrer que pour tout $n > 1$, on a

$$| x_{n+1} - x_n | \leq k^n | x_1 - x_0 |$$

(b) Déduire que la suite (x_n) ; est une suite de Cauchy.

(c) Conclure que f admet un point fixe.