

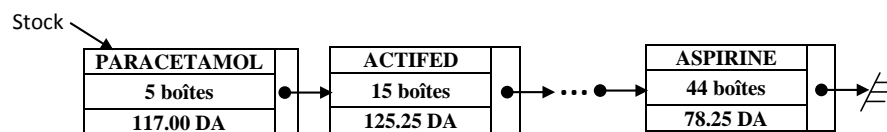
## Listes Linéaires Chainées

### Exercice 1 Algorithmes de base

- Ecrire les algorithmes de base suivants sur les listes linéaires chaînées unidirectionnelles :
  1. Suppression par valeur et par position.
  2. Inversement d'une liste :
    - En créant une nouvelle liste.
    - En Inversant le chainage dans la même liste.
  3. Tri par "sélection et permutation" et par bulles.
  4. Recherche de l'élément qui a le plus grand nombre d'occurrences.
- Ecrire les algorithmes de base suivants sur les listes linéaires chaînées bidirectionnelles :
  1. Recherche d'un élément.
  2. Suppression d'un élément.
  3. Inversement d'une liste.

**Exercice 2** Un pharmacien souhaite traiter les informations concernant son stock de médicaments par ordinateur. On vous propose de représenter ces informations sous forme de liste linéaire chaînée où chaque maillon contient le libellé d'un médicament, la quantité disponible (nombre de boîtes) et le prix unitaire.

**Exemple :**



On vous demande de :

1. Donner les structures de données nécessaires à la représentation de ce stock.
2. Ecrire la procédure **Vendre(Med, NbBoites)** permettant de retirer, si possible, 'Nb-Boites' du médicament 'Med' du stock.  
**Attention :** Il faut supprimer du stock le médicament dont la quantité atteint 0.
3. Ecrire la procédure **Acheter(Med, NbBoites, Prix)** permettant au pharmacien d'augmenter son stock par 'NbBoites' du médicament 'Med' ayant le prix unitaire 'Prix' DA. On considère qu'un médicament prend toujours le nouveau prix.
4. Ecrire la fonction **PrixStock** permettant de calculer le prix total des médicaments dans le stock.

### Exercice 3 Différence de deux listes linéaires chaînées

Soient  $L_1$  et  $L_2$  deux listes linéaires chaînées unidirectionnelles. Ecrire la procédure qui permet de construire la liste  $L = L_1 - L_2$  contenant tous les éléments appartenant à  $L_1$  et n'appartenant pas à  $L_2$ .

### Exercice 4 Liste de nombre premiers

Nous considérons la construction d'une liste des nombres premiers inférieurs ou égaux à un entier  $n$  donnée. Pour construire cette liste, on commence, dans une première phase, par y ajouter tous les entiers de 2 à  $n$  en commençant par le plus grand et en terminant par le plus petit qui se trouvera à la tête de la liste. On considère ensuite successivement les éléments de la liste dans l'ordre croissant en on supprime tous leurs multiples stricts. Ecrire cet algorithme.

### Exercice 5 Matrices creuses

Une matrice est dite creuse lorsque le nombre d'éléments nuls y figurant est très supérieur à celui des éléments non nuls. On peut représenter une matrice creuse en ne tenant compte que des éléments non nuls. Chaque ligne de la matrice est une liste linéaire chaînée ordonnée (selon le rang de la colonne) des éléments non nuls. Une table de  $N$  éléments ( $N$  étant le nombre de lignes de la matrice) donne les adresses de tête de chacune des listes. Un élément de la liste contient l'indice de la colonne et la valeur de l'élément.

1. Donner la déclaration de la structure de données nécessaire.
2. Ecrire la procédure permettant de Remplir une telle structure à partir d'une matrice  $A(M, N)$  donnée.
3. Ecrire la procédure permettant de calculer la somme de deux matrices ainsi représentées.

Sachant qu'un élément de la matrice occupe 4 octets, qu'un pointeur occupe 2 octets et qu'un indice occupe 2 octets,

1. quel est le gain d'espace quand le nombre de zéros est de  $\mu$
2. quelle valeur minimale attribuée à  $\mu$  pour que l'on puisse opter pour une telle représentation

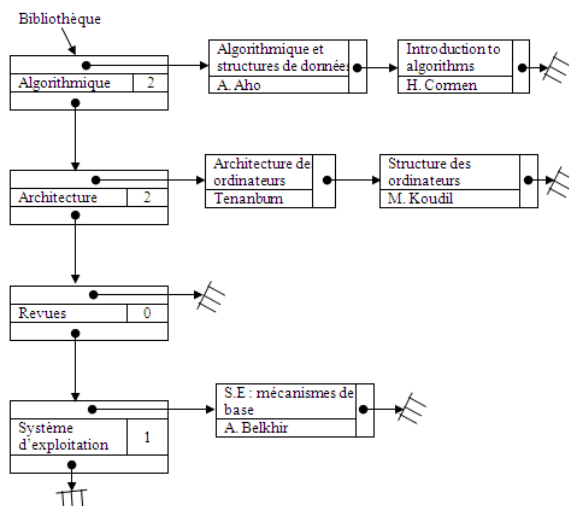
### Exercice 6 Interclassement de $n$ listes linéaires chaînées

Soit  $T$  un tableau de  $n$  listes linéaires chaînées ordonnées.

1. Transformer  $T$  en une Liste  $L$  de listes sans création de nouvelles listes
2. Donner la procédure qui affiche tous les éléments de  $L$ .
3. Créer une liste ordonnée  $LD$  contenant tous les éléments de  $T$  en parcourant parallèlement les  $n$  listes de  $T$  ( Interclassement )
4. Eliminer les doublons dans  $LD$ .

### Exercice 7 Liste de listes

Dans cet exercice, un étudiant en informatique souhaite représenter sa bibliothèque personnelle en utilisant une structure dynamique. La structure proposée est représentée dans la figure suivante :

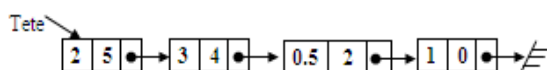


La liste verticale contient les catégories des livres avec le nombre de livres dans chacune, tandis que les listes horizontales contiennent les titres des livres avec leurs auteurs dans chaque catégorie.

1. Donner la déclaration des structures de données nécessaires à l'implémentation de cette bibliothèque ainsi que la procédure d'initialisation de ces structures.
2. Ecrire la procédure d'ajout d'une nouvelle catégorie (l'ajout se fait toujours à la fin de la liste).
3. Ecrire la procédure d'insertion d'un nouveau livre (l'ajout se fait au début de la liste)
4. Ecrire la procédure qui permette d'afficher les livres d'une catégorie donnée.
5. Ecrire la fonction qui retourne le nombre total de livres dans la bibliothèque.
6. Ecrire la procédure qui permette de supprimer une catégorie avec tous ses livres.

**Exercice 8** On souhaite représenter les polynômes par des listes linéaires chaînées, où chaque terme d'un polynôme est rangé dans un maillon de la liste contenant le degré du terme et le coefficient correspondant.

**Exemple :** Le polynôme  $2x^5 + 3x^4 + \frac{1}{2}x^2 + 1$  est représentée par la liste suivante :



On vous demande de :

1. Donner les structures de données nécessaires à la représentation des polynômes.
2. Ecrire la fonction **Valeur(P, x)** permettant de donner la valeur du polynôme P pour x.  
**Exemple :** Valeur(tete, 1) = 6.5
3. Ecrire la procédure **Dériver(P)** permettant de calculer la dérivée du polynôme P et la retourner dans P.
4. Ecrire la procédure **Intégrer(P)** permettant de calculer l'intégrale du polynôme P et le retourner dans P.
5. Ecrire la procédure **Déviations(P, x)** permettant de retourner *Vrai* si x est un point de déviation du polynôme P et *Faux* sinon.