



Ministère de l'enseignement supérieur et de la recherche scientifique  
Université Djillali BOUNAAMA - Khemis Miliana (UDBKM)  
Faculté des Sciences et de la Technologie  
Département de Mathématiques et d'Informatique



## Chapitre 4

# Listes Linéaires Chainées : Exercices corrigés

MI-L1-UEF221 : Algorithmiques et Structures de Données II

**Nouredine AZZOUZA**

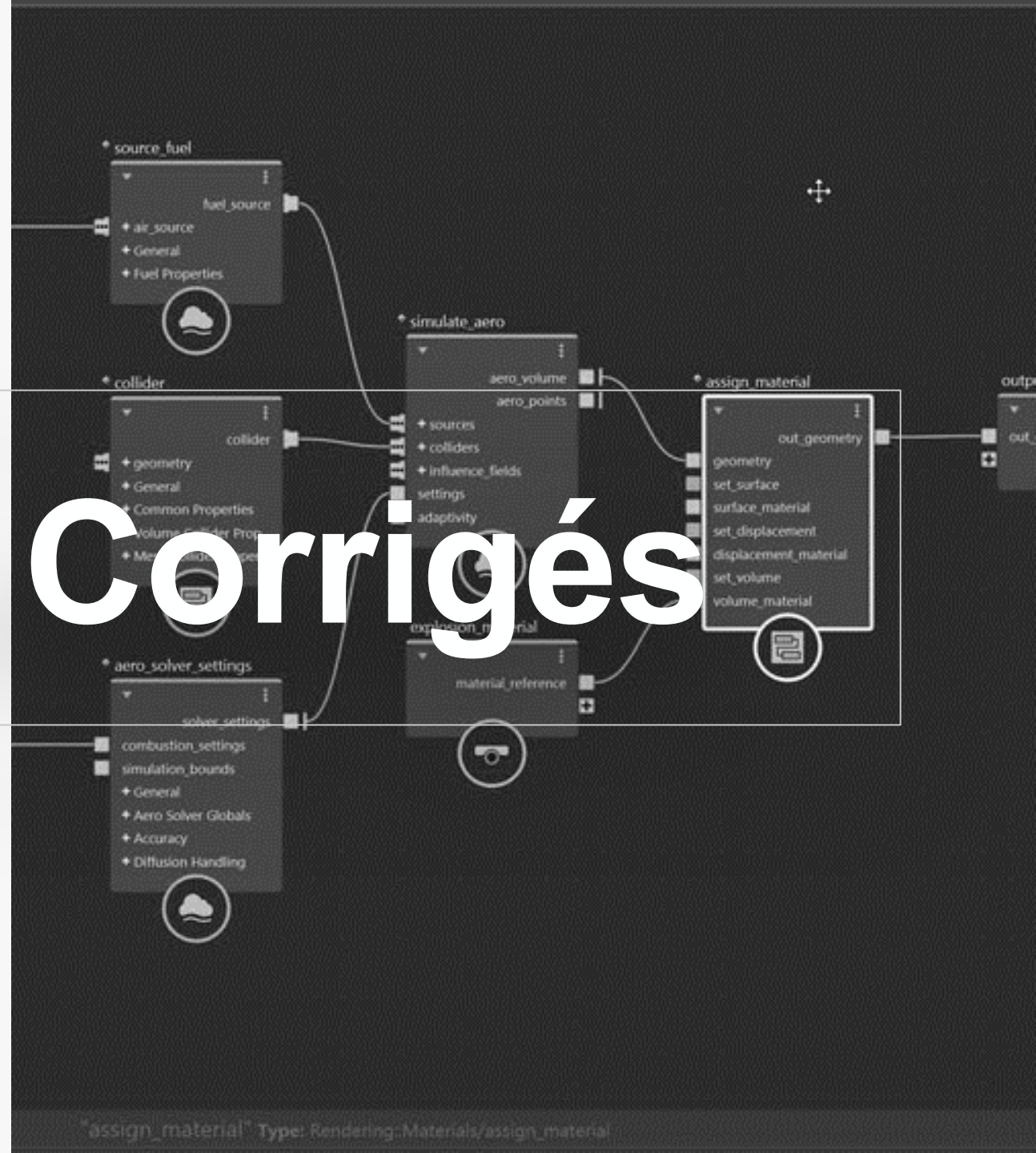
n.azzouza@univ-dbkm.dz

# Plan du Cours

**1. Exercices Corrigés**

**2. Exercices d'Examen**

# = Exercices Corrigés



## Exercices 01 : Création et parcours

✓ Ecrire un algorithme qui construit la liste L à partir de n données lues, ensuite affiche ses éléments.

✓ **Solution :**

*Algorithme CreerListe;*

*Type maillon=enregistrement*

*Val:entier;*

*Adr:pointeur(maillon)*

*Fin;*

*Var Tete, P, Q:u :*  
*pointeur(maillon);*

*i, N, V : entier ;*

*Début*

*Tete ← Nil;*

*P ← Nil;*

*Ecrire('Donner le nombre N :');*

*Lire (N);*

*//Remplir la liste*

*Pour i de1 à N faire*

*Lire(Val) ;*

*Allouer(Q) ;*

*Aff\_val(Q, val) ;*

*Aff\_adr(Q, NIL) ;*

*Si (Tete <> Nil) Alors*

*Aff\_adr(P, Q)*

*Sinon*

*Tete←Q*

*FSi;*

*P←Q;*

*FPour;*

*//Afficher les éléments de la liste*

*P ← Tete;*

*Pour i de1 à N faire*

*Ecrire(Valeur(P)) ;*

*P ← Suivant(P);*

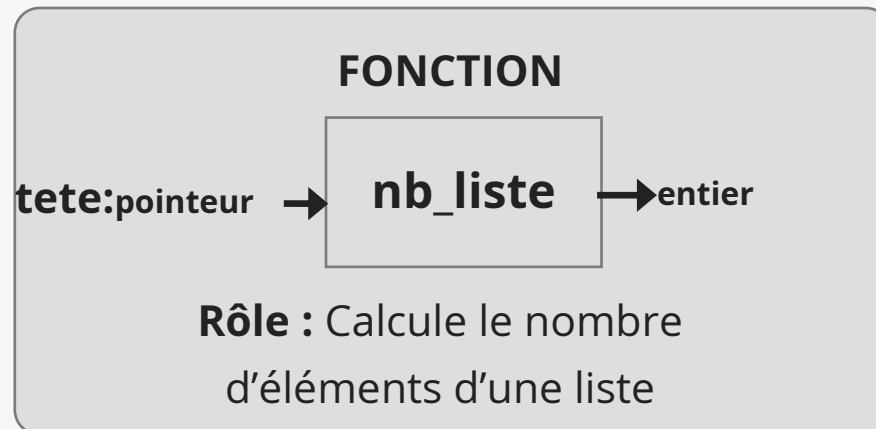
*FPour;*

*FIN.*

## Exercices 02 : Nombre d'éléments

✓ Ecrire une **fonction** qui calcule le nombre d'éléments de la liste L.

✓ **Solution :**



*Fonction*  $nb\_liste (tete: Pointeur(maillon)): entier;$

*Var*  $nbr: entier;$

$P: pointeur(maillon)$

*Début*

$P \leftarrow tete;$

$nbr \leftarrow 0;$

*TQ*  $P \langle \rangle Nil$  *Faire*

$nbr \leftarrow nbr + 1;$

$P \leftarrow Suivant (P);$

*FTQ*

$nb\_liste \leftarrow nbr ;$

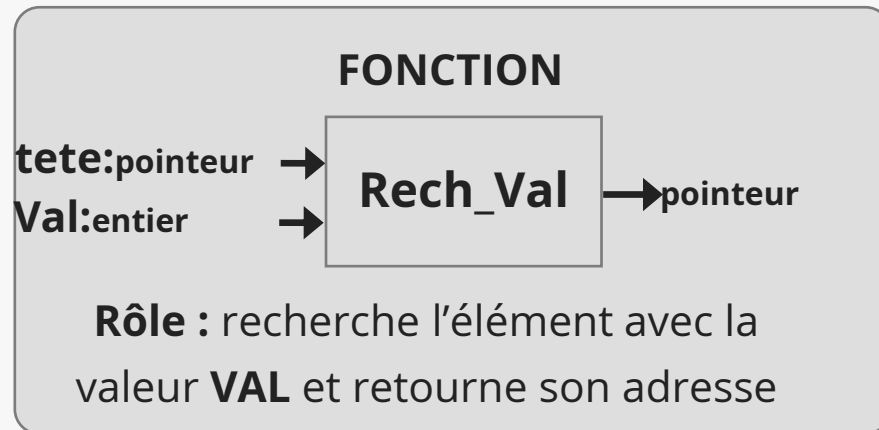
*Fin;*



## Exercices 03 : Rechercher Par valeur

- ✓ Ecrire une **fonction** qui retourne l'adresse (pointeur) de l'éléments avec la valeur VAL s'il existe sinon, retourne Nil.

- ✓ **Solution :**

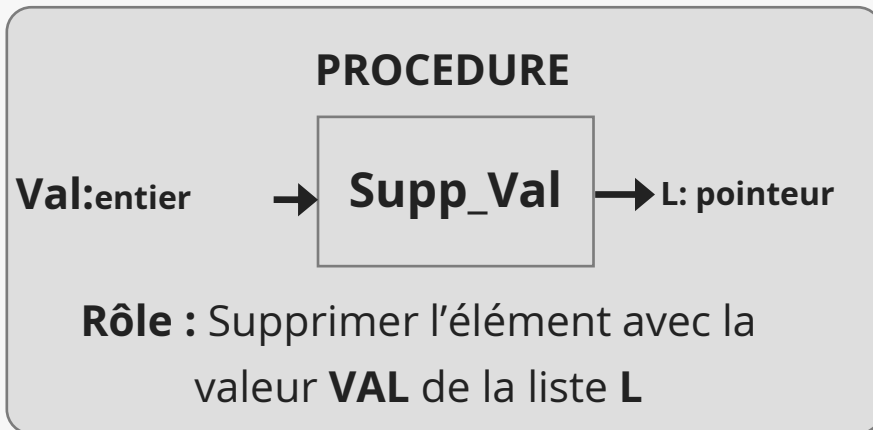


```
Fonction Rech_Val (tete: Pointeur(maillon), VAL:entier): Pointeur;  
Var P: pointeur(maillon)  
Début  
P ← tete;  
TQ P <> Nil et Valeur(P) <> VAL Faire  
P ← Suivant (P);  
FTQ  
  
Si P = Nil Alors Rech_Val ← Nil;  
Sinon Rech_Val ← P;  
Fin;
```



## Exercices 04 : Supprimer par Valeur

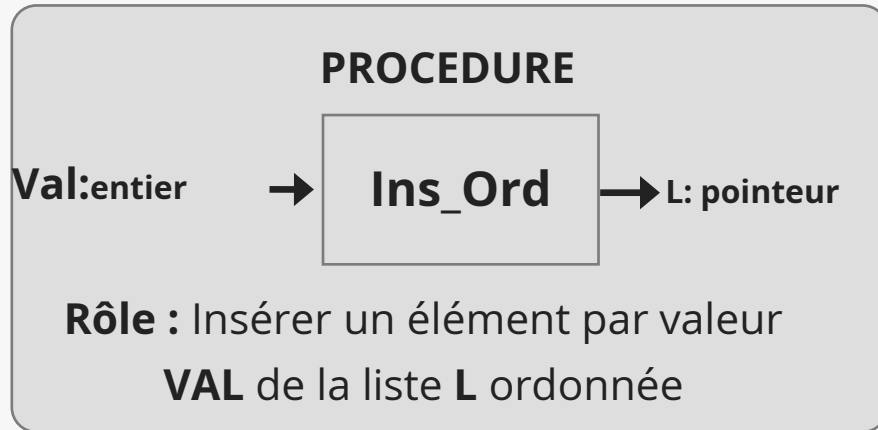
- ✓ Ecrire une **procédure** qui supprime un élément qui a la valeur **Val** dans la liste **L** s'il existe.
- ✓ **Solution :**



```
Procédure Supp_Val (VAL:entier; VAR L: Pointeur(maillon));  
Var P, Q: pointeur(maillon)  
Début  
  P ← L;  
  Q ← Nil;  
  TQ P <> Nil et Valeur(P) <> VAL Faire  
    Q ← P;  
    P ← Suivant (P);  
  FTQ  
  
  Si P <> Nil Alors  
    Si Q <> Nil Alors Aff_Adr(Q, Suivant(P));  
    Sinon L ← Suivant(P);  
    Liberer(P);  
  Fsi  
  
Fin;
```

## Exercices 05 : Insérer dans une liste ordonnées

- ✓ Ecrire une **procédure** qui supprime un élément qui a la valeur **Val** dans la liste **L** s'il existe.
- ✓ **Solution :**



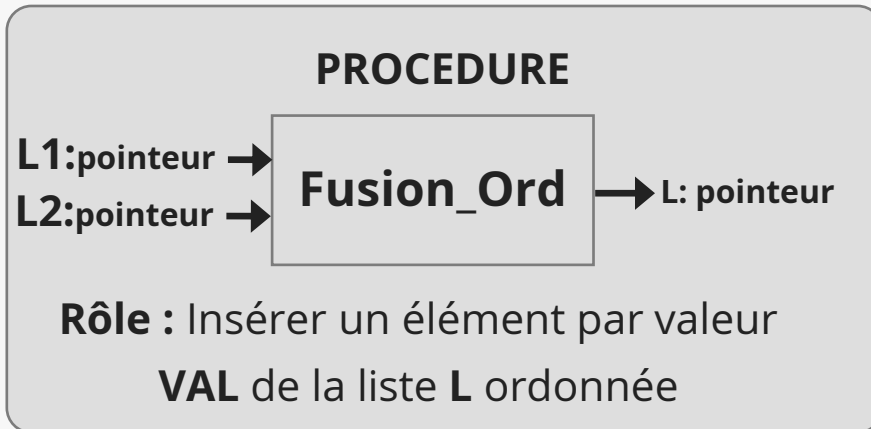
```
Procédure Ins_Ord (VAL:entier; VAR L: Pointeur(maillon));  
Var P, Q, R: pointeur(maillon)  
Début  
Allouer (P); Aff_Val(R, VAL);  
  
P ← L; Q ← Nil;  
TQ P <> Nil et Valeur(P) < VAL Faire  
    Q ← P;  
    P ← Suivant (P);  
FTQ  
  
Si Q <> Nil Alors  
    Aff_Adr(R, Suivant(Q));  
    Aff_Adr(Q, R);  
Sinon  
    Aff_Adr(R, L);  
    L ← R;  
Fsi  
  
Fin;
```



## Exercices 06 : Fusion de deux listes ordonnées

- ✓ Soit L1 et L2 deux listes linéaires chaînées ordonnées. Ecrire une **procédure** qui fusionne les deux listes pour construire une seule liste L.

✓ **Solution :**



```

Procédure Fusion_Ord (L1,L2:Pointeur(maillon); VAR L: Pointeur);
Var      P1, P2, Q : pointeur(maillon)
Début
  L ← Nil; P ← Nil;
  P1 ← L1; P2 ← L2;
  TQ P1 <> Nil et P2 <> Nil Faire
    Allouer(Q);
    Si Valeur(P1) < Valeur(P2) Alors Aff_Val(Q, Valeur(P1));
    Sinon Aff_Val(Q, Valeur(P2));

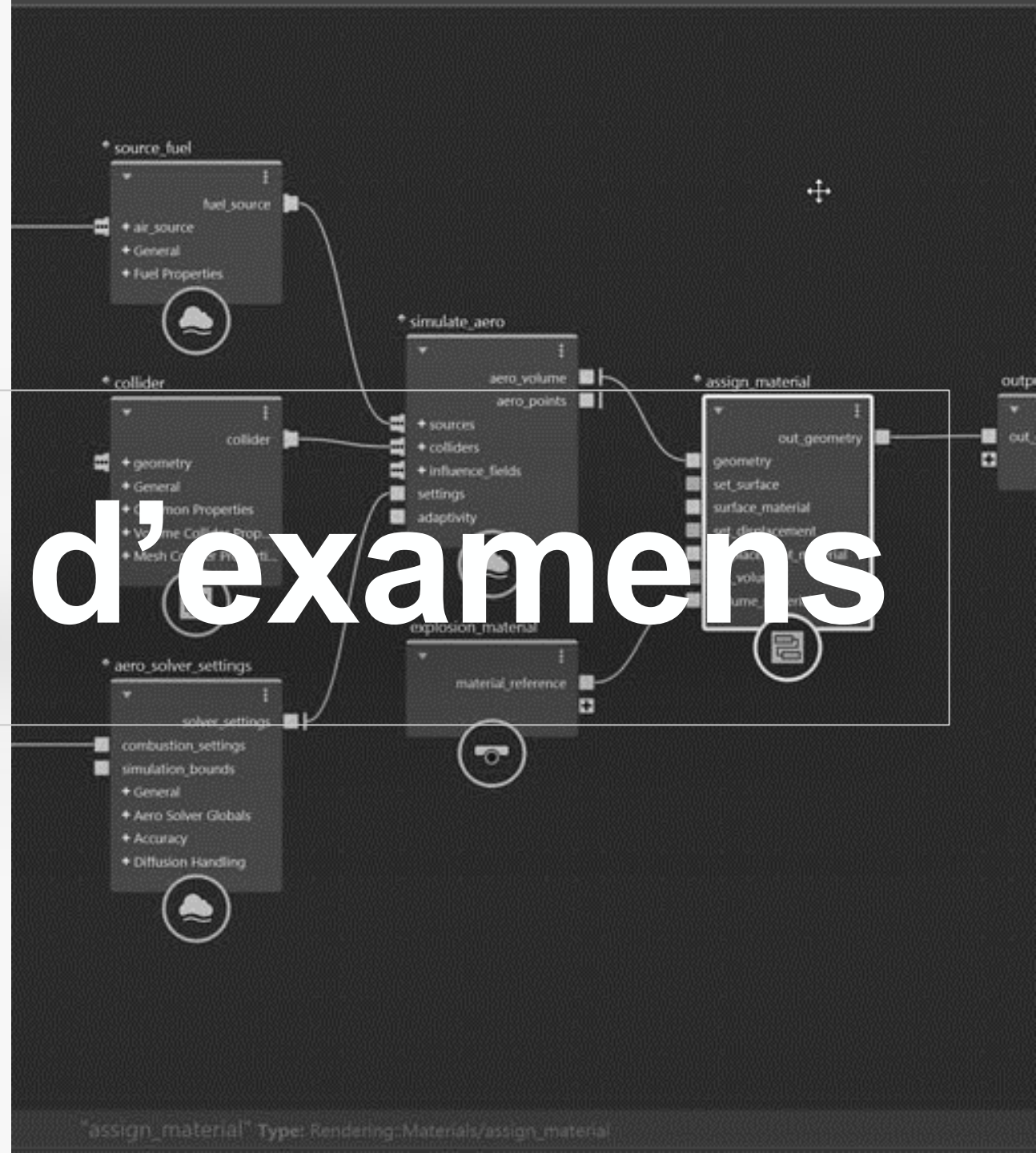
    Si L = Nil Alors L ← Q;
    Sinon Aff_Adr(P, Q);
    P ← Q;
  FTQ

  Si P1 <> Nil Alors
    Allouer(Q); Aff_Val(Q, Valeur(P1)); Aff_Adr(P, Q); P ← Q;
  Fsi
  Si P2 <> Nil Alors
    Allouer(Q); Aff_Val(Q, Valeur(P2)); Aff_Adr(P, Q); P ← Q;
  Fsi

Fin;
  
```

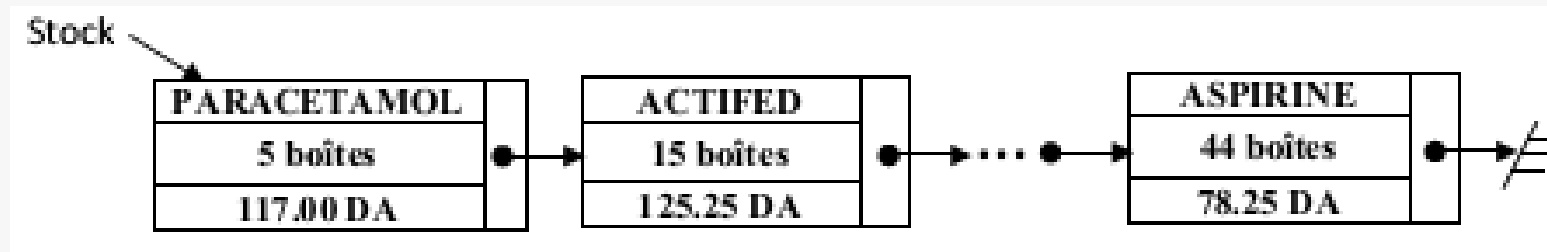
=

# Exercices d'examens



## Exercices 01 : Le pharmacien

- ✓ 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.



On vous demande de :

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

## Exercices 01 : Le pharmacien

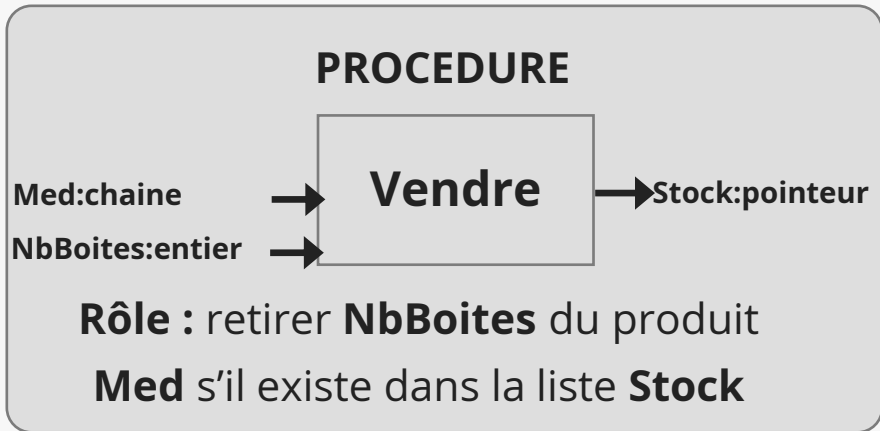
### ✓ Solution 01: Déclaration des Structures de données et remplissage de la liste des produits

```
Algorithme CreerListeParmacie;  
  Type produit=enregistrement  
    libelle:chaine;  
    Qte:entier;  
    Prix:réel;  
    Adr:pointeur(produit)  
  Fin;  
Var Tete,P,Q: pointeur(produit);  
    i, N : entier ;  
  
Début  
  Tete ← Nil;  
  P ← Nil;  
  Ecrire('Donner le nbr de produits N :');  
  Lire (N);
```

```
//Remplir la liste  
Pour i de1 à N faire  
  Allouer(Q) ;  
  Lire (Q^.libelle);  
  Lire (Q^.Qte);  
  Lire (Q^.Prix);  
  Q^.Adr ← NIL;  
Si (Tete <> Nil) Alors  
  P^.Adr ← Q;  
  
Sinon  
  Tete ← Q  
  
FSi;  
  P←Q;  
FPour;  
Fin.
```

## Exercices 01 : Le pharmacien

✓ **Solution 02: procédure Vendre (Stock, Med, NbBoites)**



### Analyse :

- Si le produit « Med » existe, on retire « NbBoites » de sa quantité
- Si sa quantité atteint 0 ou inférieur de « NbBoites », on le supprime de la liste « Stock »

```

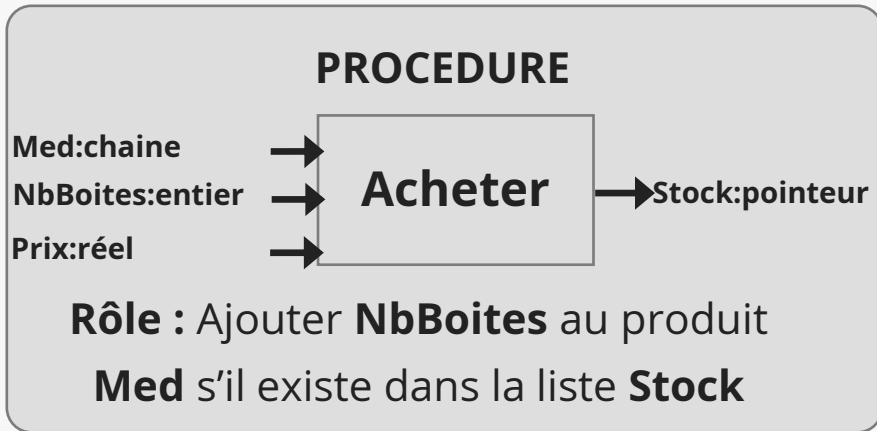
Procédure Vendre (Med:chaîne;NbBoites:entier;VAR Stock: Pointeur);
Var      P, Q: pointeur(maillon)
Début
  P ← Stock; Q ← Nil;
  TQ P <> Nil et P^.libelle <> Med Faire
    Q ← P;
    P ← Suivant (P);
  FTQ

  Si P <> Nil Alors
    Si P^.Qte > NbBoites Alors
      P^.Qte ← P^.Qte - NbBoites;
    Sinon
      Si Q <> Nil Alors Q^.Adr ← P^.Adr;
      Sinon Stock ← P^.Adr;
    Fsi
  Liberer(P);
Fsi

Fin;
```

## Exercices 01 : Le pharmacien

✓ **Solution 03: procédure Acheter (Stock, Med, NbBoites, Prix)**



### Analyse :

- Si le produit « Med » existe, on ajoute « NbBoites » à sa quantité
- S'il n'existe pas, on l'insère et on le remplit avec « NbBoites » et « prix »

```
Procédure Acheter(Med:chaîne; NbBoites:entier; Prix : Reel;
                  VAR Stock: Pointeur(produit) );
```

```
Var P, Q, R: pointeur(maillon)
```

```
Début
```

```
P ← Stock; Q ← Nil;
```

```
TQ P <> Nil et P^.libelle <> Med Faire
```

```
Q ← P;
```

```
P ← Suivant (P);
```

```
FTQ
```

```
Si P <> Nil Alors
```

```
P^.Qte ← P^.Qte + NbBoites;
```

```
Sinon
```

```
Allouer(R);
```

```
R^.libelle ← Med; R^.Qte ← NbBoites; R^.Prix ← Prix;
```

```
Si Stock = Nil Alors Stock ← R;
```

```
Sinon Q^.Adr ← R;
```

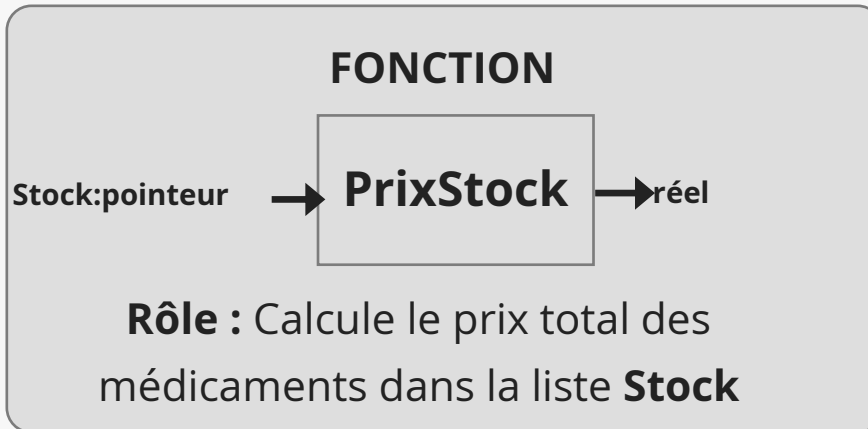
```
Fsi
```

```
Fsi
```

```
Fin;
```

## Exercices 01 : Le pharmacien

### ✓ Solution 04: fonction PrixStock (Stock)



### Analyse :

- Parcourir tous les produits de la liste « stock » et à chaque fois on :
  - On multiplie la quantité du produit par son prix
  - Et on ajoute le résultat à la somme globale

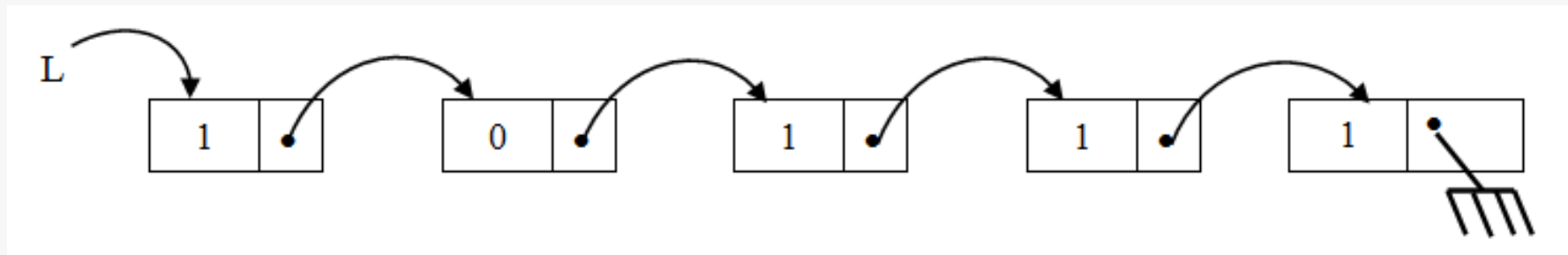
```
Fonction PrixStock(Stock: Pointeur(produit) ): réel;  
Var      P : pointeur(maillon);  
          S: Réel;  
  
Début  
  P ← Stock;  
  S ← 0;  
  TQ P <> Nil Faire  
    S ← S + P^.Qte * P^.Prix;  
    P ← Suivant (P);  
FTQ  
  
  PrixStock ← S;  
  
Fin;
```

## Exercices 02 : Conversion Décimal-Binaire

Soit N un nombre entier écrit sous la base 10.

- Ecrire un algorithme qui permet de convertir N en binaire et le stocker dans une liste linéaire chaînée chiffre par chiffre.

**Exemple** : Si  $N = 29$  ; alors  $(29)_{10} = (11101)_2$





## Exercices 02 : Conversion Décimal-Binaire

✓ **Solution :**

```
Algorithme ConvertDecimalBinaire;  
  Type maillon=enregistrement  
    Val:entier;;  
    Adr:pointeur(maillon)  
  Fin;  
  Var L,P,Q: pointeur(produit);  
    N, R : entier ;  
Début  
  L ← Nil;  
  P ← Nil;  
  Ecrire('Donner le nombre N :');  
  Lire (N);
```

```
//Convertir N et remplir la liste  
Répéter  
  R ← N MOD 2;  
  N ← N DIV 2;  
  Allouer(Q) ;  
  Aff_Val(Q, R);  
  Aff_Adr(Q, Nil);  
  Si (Tete <> Nil) Alors Aff_Adr(P, Q);  
  Sinon L ← Q;  
  FSi;  
  P←Q;  
Jusqu'à (N = 0);  
  
Fin.
```