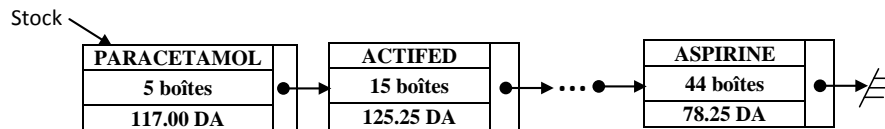


## Examen

### Exercice 1 LLCs (8 pts : 1 + 3 + 2.5 + 1.5)

Un pharmacien souhaite traiter les informations concernant son stock de médicaments par ordinateur. On vous propose de représenter ces informations sous forme de liste 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 :

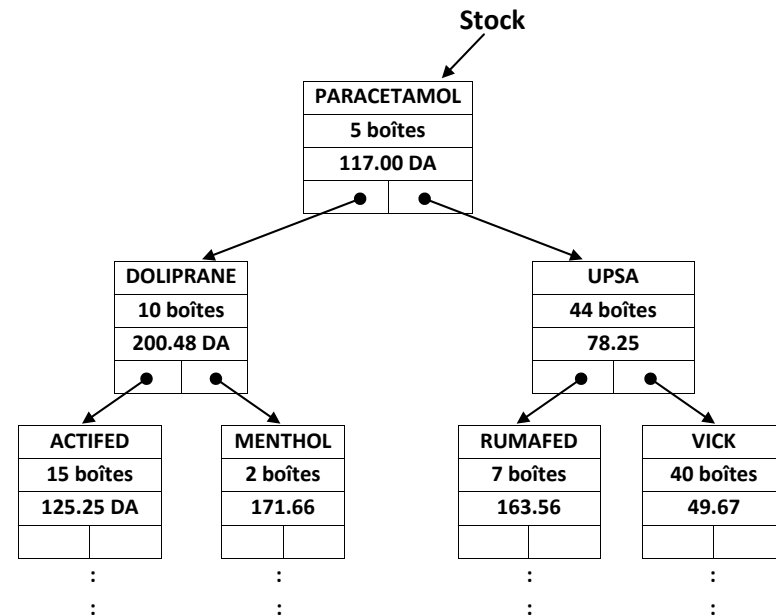
- Donner les structures de données nécessaires à la représentation de ce stock.
- Ecrire la procédure **Vendre(Med, NbBoites)** permettant de retirer, si possible, 'NbBoites' du médicament 'Med' du stock.  
**Attention :** Il faut supprimer du stock le médicament dont la quantité atteint 0.

- Ecrire la procédure **Acheter(Med, NbBoites, Prix)** permettant au pharmacien d'alimenter son stock par 'NbBoites' du médicament 'Med' ayant le prix unitaire 'Prix' DA. On considère qu'un médicament prene toujours le nouveau prix.
- Ecrire la fonction **PrixStock** permettant de calculer le prix total des médicaments dans le stock.

### Exercice 2 ARB(9 pts : 1 + 3 + 3 + 2)

Le pharmacien, appréciant la méthode des LLCs, a remarqué rapidement la lenteur des procédures Vendre et Acheter et il vous demande une solution plus rapide. On vous propose de réécrire la solution précédente en utilisant un arbre de recherche binaire basé sur l'ordre alphabétique des libellés des médicaments.

**Exemple :**

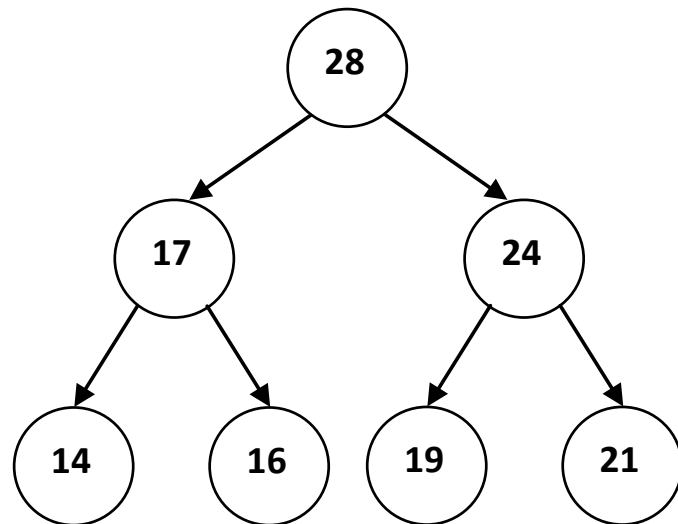


On vous demande de :

1. Donner les structures de données nécessaires à la représentation de ce stock.
2. Réécrire la procédure **Vendre** décrite dans l'exercice précédent sur la nouvelle structure.  
**NB** : On donne la procédure **Nettoyer (Stock)** permettant de supprimer du stock tous les médicaments ayant une quantité nulle.
3. Réécrire la procédure **Acheter** décrite dans l'exercice précédent sur la nouvelle structure.
4. Réécrire la fonction **PrixStock** décrite dans l'exercice précédent sur la nouvelle structure.

**Exercice 3** (3 pts : 1 + 1 + 1)

Soit l'arbre binaire suivant :



1. Que représente cette structure : un ARB ou un tas ?

2. Donner cette structure après l'ajout de 40 puis 22.
3. Donner cette structure après la suppression de 19 puis 28.

★★★ Bonne chance ★★★

A.Djeffal

## Corrigé type

### Exercice 1 : Listes linéaires chaînées

#### 1. Structures de données (1 pt)

```
Type TMaillon = Structure  
  Libellé : chaine ;  
  Quantité : entier ;  
  PrixUnit : réel ;  
  Suivant : Pointeur(TMaillon) ;  
Fin ;  
Var Stock : Pointeur(TMaillon) ;
```

```
Procédure Vendre( Med : chaine, NbBoites : entier);  
Var P,PP : Pointeur(TMaillon) ;  
Début  
  P ← Stock ;  
  PP ← Nil ;  
  Tant que ((P ≠ Nil et Libellé(P) ≠ Med) faire  
    | PP ← P ;  
    | P ← Suivant(P) ;  
  Fin TQ ;  
  Si (P ≠ Nil) Alors  
    | Si (Quantité(P) < NbBoites) Alors  
      | Ecrire('Quantité insuffisantes')  
    | Sinon  
      | Aff_Quantité(P,Quantité(P)-NbBoites) ;  
      | Si (Quantité(P) = 0) Alors  
        | Si (PP=Nil) Alors  
          | Stock ← Suivant(Stock)  
        | Sinon  
          | Aff_Adr(PP,Suivant(P))  
        | Fin Si ;  
      | Libérer(P) ;  
      | Fin Si ;  
    | Fin Si ;  
  | Sinon  
    | Ecrire('Ce médicament n'existe pas') ;  
  | Fin Si ;  
Fin ;
```

#### 2. Procédure Vendre(Med, NbBoîtes) (3 pts)

3. Procédure Acheter(Med, NbBoites, PrixUnit) (2.5 pts)

```
Procédure Acheter( Med : chaîne, NbBoites : entier, PrixUnit : réel);
Var P : Pointeur(TMaignon);
Début
  P ← Stock;
  Tant que ((P ≠ Nil et Libellé(P) ≠ Med) faire
    | P ← Suivant(P);
  Fin TQ;
  Si (P ≠ Nil) Alors
    | Aff_Quantité(P,Quantité(P)+NbBoites);
    | Aff_PrixUnit(P,PrixUnit);
  Sinon
    | Allouer(P);
    | Aff_Libellé(P,Med);
    | Aff_Quantité(P,NbBoites);
    | Aff_PrixUnit(P,PrixUnit);
    | Aff_Adr(P,Stock);
    | Stock ← P;
  Fin Si;
Fin;
```

```
Fonction PrixStock() : réel;
Var P : Pointeur(TMaignon);
    S : réel
Début
  P ← Stock;
  S ← 0;
  Tant que ((P ≠ Nil) faire
    | S ← S + Quantité(P) × PrixUnit(P);
    | P ← Suivant(P);
  Fin TQ;
  PrixStock ← S;
Fin;
```

**Exercice 2 : Arbres de recherche binaires**

1. Structures de données (1 pt)

```
Type TNoeud = Structure
  Libellé : chaîne;
  Quantité : entier;
  PrixUnit : réel;
  FG,FD : Pointeur(TNoeud);
Fin;
Var Stock : Pointeur(TNoeud);
```

4. La fonction PrixStock (1.5 pt)

2. Procédure Vendre(Med, NbBoîtes) (3 pts)

```
Procédure Vendre( Med : chaîne, NbBoites : entier);  
Var P,PP : Pointeur(TNoeud);  
Début  
  P ← Stock;  
  PP ← Nil;  
  Tant que ((P ≠ Nil et Libellé(P) ≠ Med) faire  
    PP ← P;  
    Si (Libellé(P) > Med) Alors  
      | P ← FG(P)  
    Sinon  
      | P ← FD(P)  
    Fin Si;  
  Fin TQ;  
  Si (P ≠ Nil) Alors  
    Si (Quantité(P) < NbBoites) Alors  
      | Ecrire('Quantité insuffisante')  
    Sinon  
      | Aff_Quantité(P,Quantité(P)-NbBoites);  
      | Si (Quantité(P) = 0) Alors  
        | Nettoyer(P)  
      | Fin Si;  
    Fin Si;  
  Sinon  
    | Ecrire('Ce médicament n'existe pas');  
  Fin Si;  
Fin;
```

3. Procédure Acheter(Med, NbBoites, PrixUnit) (3 pts)

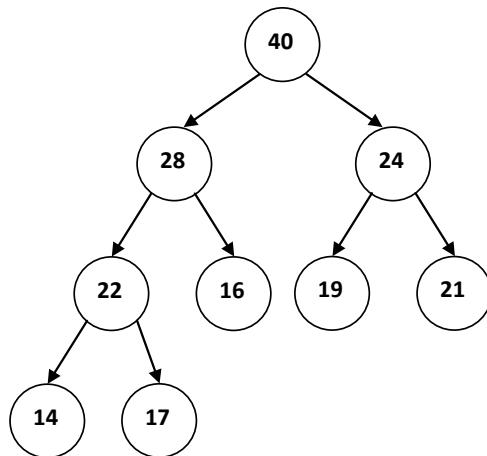
```
Procédure Acheter( Med : chaîne, NbBoites : entier, PrixU-  
nit : réel);  
Var P,PP : Pointeur(TNoeud);  
Début  
  P ← Stock; PP ← Nil;  
  Tant que ((P ≠ Nil et Libellé(P) ≠ Med) faire  
    PP ← P;  
    Si (Libellé(P) > Med) Alors  
      | P ← FG(P)  
    Sinon  
      | P ← FD(P)  
    Fin Si;  
  Fin TQ;  
  Si (P ≠ Nil) Alors  
    | Aff_Quantité(P,Quantité(P)+NbBoites);  
    | Aff_PrixUnit(P,PrixUnit);  
  Sinon  
    | Allouer(P);  
    | Aff_Libellé(P,Med);  
    | Aff_Quantité(P,NbBoites);  
    | Aff_PrixUnit(P,PrixUnit);  
    | Aff_FG(P,Nil); Aff_FD(P,Nil);  
    Si (PP=Nil) Alors  
      | Stock ← P  
    Sinon  
      | Si (Libellé(P) > Med) Alors  
        | Aff_FG(PP,P)  
      | Sinon  
        | Aff_FD(PP,P)  
      | Fin Si;  
    Fin Si;  
  Fin Si;  
Fin;
```

4. La fonction PrixStock (2 pt)

```
Fonction PrixStock( P : Pointeur(TNoeud)) : réel;  
Début  
  Si (P=Nil) Alors  
    PrixStock ← 0  
  Sinon  
    PrixStock ← Quantité(P) × PrixUnit(P) + Prix-  
    Stock(FG(P)) + PrixStock(FD(P));  
  Fin Si;  
Fin;
```

### Exercice 3

1. Cette structure représente un tas. (1 pt)
2. Le tas après l'ajout de 40 puis 22. (1 pt)



3. Impossible de supprimer 19 du tas (0.5 pt). Après le retrait de 28, le tas devient (0.5 pt) :

