Université Ibn Tofail **A.U. 2017/2018**

Faculté des Sciences **Filière : SMI/SMA**

Département d'Informatique **Semestre 4**

Kénitra **Structures de Données.**

Examen (1h30)

Exercice 1 :

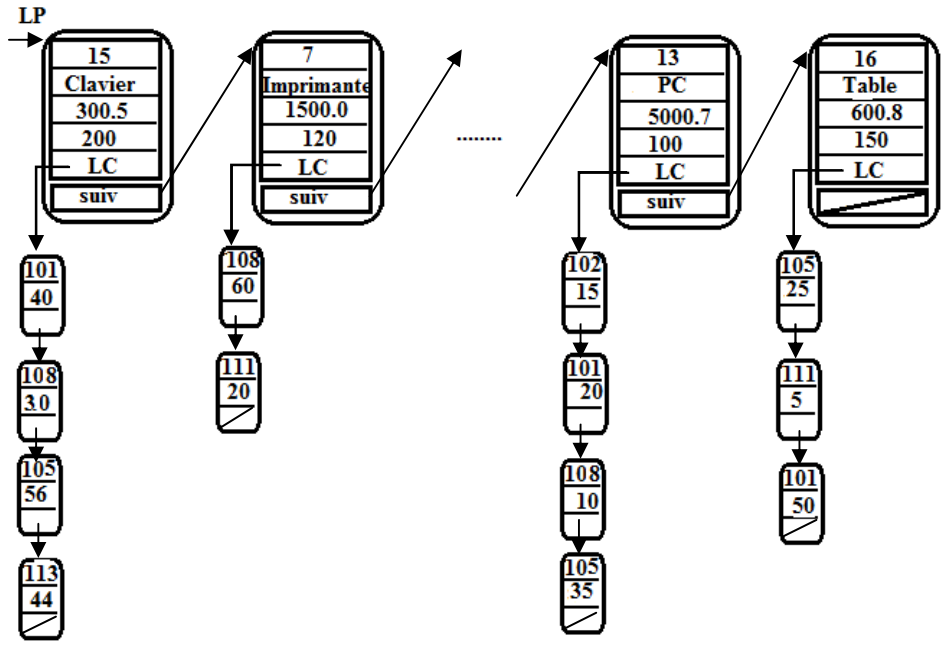
1. Construire l'arbre binaire de recherche obtenu en insérant les éléments de la liste suivante dans leur ordre d'arrivée: 14, 23, 4, 9, 17, 11, 28, 16, 3, 7.
2. Dessiner l'arbre après chaque suppression des clés : 17, 23, 14
3. Définir la structure de données **ArBin** pour représenter les arbres binaires d'entiers.

Ecrire les fonctions suivantes :

4. Fonction **nbPairs** (ArBin *R) qui retourne le nombre de clés (valeurs) paires dans l'arbre binaire de racine R.
5. Fonction **nbSup** (ArBin *R , int v) qui retourne le nombre de clés supérieures à v dans l'arbre binaire de recherche de racine R.
6. Fonction **fusion**(ArBin *R1, ArBin *R2) qui prend deux arbres binaires de recherches de racines R1, R2 et retourne la racine d'un arbre binaire de recherche contenant les deux. Utilisez les deux fonctions **suppr**(R, v) et **inser**(R, v) qui permettent respectivement la suppression et l'insertion de clé v dans l'arbre binaire de racine R.

Exercice 2 :

Un ensemble des produits est représenté par une liste **LP**. Chaque produit est caractérisé par : un identificateur **idp** (entier), un **nom** (Chaîne de 10 caractères), un prix unitaire **pu** (réel), une quantité du stock **qs** (entier) et une liste des clients **LC** demandant le produit. Chaque client de la liste **LC** est décrit par un identificateur **idc** (entier) et une quantité demandée **qdc** (entier).



La gestion des listes des produits **LP** et des clients **LC** doit respecter les contraintes suivantes :

- Un produit n'apparaît qu'une seule fois dans la liste des produits **LP**.
- Un client peut demander plusieurs produits, mais ne peut demander qu'une seule fois le même produit.
- Après une demande d'un produit par un client, ce client est ajouté à la liste des clients **LC** demandant ce produit si la quantité demandée est inférieure à la quantité restante en stock.

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

- **creerListeProd**() : permet de créer une structure vide.
- **rechProd**(LP, idp) : permet de vérifier l'existence d'un produit d'identificateur **idp** dans la liste **LP**. Elle retourne l'adresse du maillon contenant le produit s'il existe sinon elle retourne NULL.
- **ajoutProd**(LP, idp, nomp, prix, qt) : permet d'alimenter la structure **LP** par **qt** unités du produit '**nomp**' ayant le prix unitaire '**prix**' et d'identificateur **idp**. Si le produit existe déjà dans la structure, l'opération d'ajout augmente sa quantité par **qt**. L'ajout sera au début de la liste.
- **supprimProd**(LP, idp) : supprime un produit, s'il existe, de la structure **LP** dont l'identificateur est passé en paramètre.
- **demandeClientProd**(PR, idp, idc, qdc) : ajoute un client (**idc**, **qdc**) au début de la liste des clients **CL** demandant le produit d'identificateur **idp** si la quantité demandée est inférieure à la quantité restante en stock, sinon affiche le message "quantité du stock insuffisante ".
- **factureClient**(LP, idc) : Calcule le prix total des produits demandés par un client d'identificateur **idc**.

1. Donner la spécification abstraite de la structure de données **LProduit**.
2. On veut représenter les structures **LClient** et **LProduit** des listes des clients et de produits respectivement sous forme de listes simplement chaînées. Définir les différentes structures de données utiles.
3. Implémenter les opérations précédentes.