

**Exercice 1 :**

Une société commerciale souhaite informatiser son système de gestion de ventes. Pour cela, son directeur vous fait appel afin que vous lui proposiez un algorithme qui sera traduit en un programme C.

La facture d'une transaction de vente contient les informations suivantes : un numéro de transaction, le nom du client, le label du produit vendu, la quantité du produit vendu, le montant de la transaction, la date de la transaction (jour, mois, et année).

- Quelle est la structure de données adéquate pour modéliser une facture de vente ? Justifiez votre choix.

À la fin de l'année, le directeur souhaite consulter le fichier de factures stocké dans son ordinateur pour avoir quelques informations à propos de ses ventes. Par exemple, pour un produit **P1** : quelle est la quantité totale vendue du produit ? et quel est le client qui a acheté la plus grande quantité du produit, toute en indiquant le jour et le mois de la transaction ?

- Justifiez l'utilisation des fichiers dans un tel algorithme.
- Proposez un algorithme au directeur qui lui permettra d'avoir les informations souhaitées.

**Exercice 2 :**

Soit un code, en langage C, enregistré dans un fichier 'Code.txt'. Chaque instruction du code est écrite sur une ligne distincte, et un commentaire est inséré juste en dessus de chacune de ces instructions. On suppose que le programmeur du code n'a utilisé que le symbole '//' pour désigner un commentaire.

- Ecrivez un algorithme qui supprime tous les commentaires du code.

## Corrigé

### Exercice 1 :

- 1- Un enregistrement. Car, les informations d'une facture ne sont pas du même type. Les enregistrements permettent de stocker des champs de types différents pour former une entité complète.
- 2- Les transactions de vente doivent être stockées d'une manière permanente dans l'ordinateur. Ce qui justifie l'utilisation d'un fichier.
- 3- L'algorithme :

**Algorithme** exo1 ;

**Type** dateT = enregistrement

Jour : 1...31 ;

Mois : 1..12 ;

Annee : Entier ;

**Fin ;**

**Type** facture = enregistrement

numero : entier ;

nom, label : chaîne de caractères ;

label, quantite: réel ;

date : dateT ;

**Fin ;**

**Var**

F : fichier de facture ;

T : facture ;

Total, Commande, J, M : entier ;

Nom, P1 : chaîne de caractères ;

**Début**

Associer (F, 'Text.txt') ;

Relire (F) ;

Total  $\leftarrow$  0 ;

Ecrire ('Veuillez introduire le label du produit') ;

Lire (P1) ;

**Tant que** non FDF(F) **faire**

    Lire (F, T) ;

**Si** (T.label = P1) **alors**

        Total = Total + T.quantite ;

**Fin Si ;**

**Fin tant que ;**

Ecrire ('La quantité totale vendue du produit P1 est', Total) ;

Fermer (F) ;

Relire (F) ;

Commande  $\leftarrow$  0 ;

**Tant que** (non FDF(F)) **faire**

Lire (F,T) ;

**Si** (T.label = 'P1') et (T.quantite > Commande) **alors**

Nom  $\leftarrow$  T.nom ;

Commande  $\leftarrow$  T.quantite ;

J  $\leftarrow$  T.date.jour ;

M  $\leftarrow$  T.date.mois ;

**Fin si** ;

**Fin tant que** ;

Fermer(F) ;

Ecrire ('Le client', Nom, 'a pris la quantité', Commande, ' le', J, '/', M, '/2022') ;

**Fin.**

## **Exercice 2 :**

**Algorithme** exo2 ;

**Var**

F,G : texte ;

S : chaîne de caractères ;

**Début**

Associer (F, 'Code.txt') ;

Associer (G, 'CodeSC.txt') ;

Relire (F) ;

Réécrire (G) ;

**Tant que** (non FDF(F)) **faire**

Lire (F,S) ;

Si (S[1]  $\diamond$  '/') alors

Ecrire (G, S) ;

**Fin Si** ;

**Fin Tant que** ;

Fermer (F) ;

Fermer (G) ;

**Fin.**