

Exercice 1:

Nous souhaitons écrire un algorithme pour gérer les clients d'une banque. Un client est caractérisé par les informations suivantes : Nom et Prénom (30 caractère), le numéro du client (séquentiel), le solde (réel), et le numéro de compte. Le numéro de compte est une chaîne de caractère composée des trois premières lettres du nom du client suivies du numéro séquentiel.

Travail à faire :

1. Définir le type *client* permettant de sauvegarder les informations des clients de la banque.
2. Ecrire une fonction *Num_Compte* qui renvoie le numéro de compte d'un client généré à partir de son nom et de son numéro séquentiel.
3. Ecrire une procédure *Lire_client* permettant de lire les informations relatives à une variable de type client passée en paramètre.
4. Ecrire un sous-programme *Débiter* permettant de débiter une somme d'argent du compte d'un client. Le numéro de compte du client et la somme à débiter sont passés en paramètre.
5. En utilisant les sous-programmes définis précédemment, écrivez l'algorithme permettant de sauvegarder les informations de *N* clients de la banque dans un tableau. L'algorithme doit ensuite débiter le compte d'un client dont le numéro de compte et la somme à débiter sont donnés par l'utilisateur. L'algorithme affiche le message '*solde insuffisant*' lorsque le client n'a pas assez d'argent dans son compte pour effectuer la transaction.

Exercice 2:

Le gérant d'un hôtel souhaite informatiser la gestion des chambres. Pour cela on utilise un tableau d'enregistrements. Une chambre est caractérisée par: un numéro (entier), le type (simple, double, suite), le prix (réel), disponible (vrai/faux). La date de la dernière rénovation (jour: 1..31, mois: 1..12, année: entier)

Travail à faire:

1. Définir les types nécessaires pour la réalisation de l'algorithme, à savoir : *date* et *chambre*
2. Ecrire une fonction booléenne *compare_date(d1,d2)* pour comparer deux dates d1 et d2. La fonction renvoie vrai lorsque d1 est récente par rapport à d2 et renvoie faux sinon.
3. Ecrire l'algorithme principal permettant de réaliser les traitements suivants :
 - a. Sauvegarder les informations des chambres dans un tableau. le nombre de chambre est introduit par l'utilisateur.
 - b. Afficher le numéro de la chambre la plus récente.
 - c. Allouer une chambre au choix de l'utilisateur, le numéro de chambre est lu au clavier. L'algorithme doit d'abord afficher les chambres disponibles pour donner le choix à l'utilisateur.