

1) On souhaite stocker sur disque dur ou disquette les opérations concernant des comptes bancaires. Chaque opération constitue une ligne de la forme :

Numéro_de_compte Date Montant Libellé.

Écrire un algorithme permettant de saisir un nombre indéterminé d'opérations et de les stocker dans un fichier. On considère que l'utilisateur ne commet pas d'erreur de saisie.

```
Algorithme Operation_Ecrire

Var fich : fichier texte
  num : entier
  date, libelle : chaîne
  montant : réel

Début
  ' Ouverture du fichier
  Ouvrir(fich)
  ' Lecture premier numéro de compte
  Lire(num)
  ' Traitement tant que les numéros de compte saisis sont positifs
  Tant que num>0 faire
    ' Lecture des autres données
    Lire(date, montant, libelle)
    ' Écriture dans le fichier
    Écrire(fich, num, date, montant, libelle)
    ' Lecture du numéro de compte suivant
    Lire(num)
  Fin tant que
  ' Fermeture du fichier
  Fermer(fich)
Fin
```

2) Écrire un algorithme permettant de relire et d'afficher les données stockées dans le fichier précédent.

```
Algorithme Operation_Lire

Var fich : fichier texte
  num : entier
  date, libelle : chaîne
  montant : réel

Début
  ' Ouverture du fichier
  Ouvrir(fich)
  ' Lecture du fichier jusqu'à la fin
  Tant que non EOF(fich) faire
    ' Lecture des données
    Lire(fich, num, date, montant, libelle)
    ' Affichage à l'écran
    Afficher(num, date, montant, libelle)
  Fin tant que
  ' Fermeture du fichier
```

```
  Fermer(fich)
Fin
```

3) Écrire un algorithme permettant d'ajouter les données du fichier précédent aux données d'un fichier préexistant, non vide, de même structure.

```
Algorithme Operation_Lire_Ajouter

Var fich_ancien, fich : fichier texte
  num : entier
  date, libelle : chaîne
  credeb : car
  montant : réel

Début
  ' Ouverture des fichiers
  Ouvrir(fich) ' en lecture
  Ouvrir(fich_ancien) ' en ajout
  ' Lecture du nouveau fichier jusqu'à la fin
  Tant que non EOF(fich) faire
    ' Lecture des données
    Lire(fich, num, date, montant, libelle)
    ' Écriture des données
    Ajouter(fich_ancien, num, date, montant, libelle)
  Fin tant que
  ' Fermeture des fichiers
  Fermer(fich_ancien)
  Fermer(fich)
Fin
```

4) On désire mettre en place un répertoire de personnes afin de pouvoir leur adresser ultérieurement des mailings (papiers ou électroniques). Les renseignements à conserver pour chaque personne sont : nom, prénom, adresse postale (numéro, rue, code postal, ville, pays), numéro de téléphone, numéro de télécopie, et adresse mél. Déclarer un type enregistrement « Adresse », un type enregistrement « Personne » et un type « Répertoire » qui est un tableau de 1000 personnes. Déclarer une variable *rep* de type Répertoire.

```
Type Adresse = Enregistrement
  Num : entier
  Rue : chaîne
  Cp : entier
  Ville : chaîne
  Pays : chaîne
Fin

Type Personne = Enregistrement
  Nom : chaîne
  Prénom : chaîne
  Adr : Adresse
  Tel : entier
  Fax : entier
  Mél : chaîne
Fin

Type Répertoire = Tableau (1..1000) de Personne

Var rep : Répertoire
```

- 5) On souhaite calculer, d'après un capital de départ et un taux de rémunération annuel, le capital obtenu après X années. Écrire une procédure prenant en paramètres le capital de départ, le taux et le nombre d'années et permettant d'afficher à l'écran le capital final. Donner un ou deux exemples d'appels à cette procédure.

```
Procédure Affiche_Capital(cap_depart, taux : réel, N : entier)

Var cap : réel
    i : entier

Début
    cap:=cap_depart
    Pour i de 1 à N faire
        cap:=cap*(1+taux)
    Fin pour
    Afficher('Capital final',cap)
Fin

' Exemples d'appels à la procédure Affiche_Capital

Algorithme Capital

Var K, tx : réel
    N : entier

Début
    Affiche_Capital(500, 0.045, 3)
    Lire(K, tx, N)
    Affiche_Capital(K, tx, N)
Fin
```

- 6) On souhaite calculer le salaire des commerciaux d'une entreprise, en se basant sur leur salaire fixe et leur chiffre d'affaire. Si le chiffre d'affaire est supérieur à 50.000 F, le fixe est augmenté de 25 % et de seulement 5 % sinon. Écrire une fonction prenant en paramètres le fixe et le chiffre et renvoyant le salaire total. Donner un ou deux exemples d'appels à cette fonction.

```
Fonction Commission(fixe, chiffre : réel) : réel

Début
    Si chiffre>50000 alors
        Commission:=fixe*1,25
    Sinon
        Commission:=fixe*1,05
    Fin si
Fin

' Exemples d'appels à la fonction Commission

Algorithme Calcul_com

Var com1, com2 : réel

Début
    com1:=Commission(10000,35000)    ' com1 sera égale à 10500
    com2:=Commission(9000,60000)    ' com2 sera égale à 11250
Fin
```

Note de rappel sur les modes de passage des paramètres d'une procédure

Par valeur	Par adresse
La valeur des paramètres peut être utilisée dans la procédure, mais pas modifiée.	La valeur des paramètres peut être modifiée. Les paramètres passés par adresse sont précédés du mot clé Var.

Paramètres en entrée

Paramètre en sortie ou en entrée/sortie

- 7) Réécrire la procédure de calcul de capital de façon à modifier la valeur du capital de départ (paramètre en entrée/sortie). Le taux et le nombre d'annuités restent des paramètres en entrée. Ne pas afficher le résultat du calcul. Donner un exemple d'appel à cette procédure.

```
Procédure Calcul_Capital(var cap : réel, taux : réel, n : entier)

Var i : entier

Début
    Pour i de 1 à n faire
        cap:=cap*(1+taux)
    Fin pour
Fin

' Exemple d'appel à la procédure Calcul_Capital

Algorithme Capital2

Var K, tx : réel

Début
    K=1000
    Tx=0.05
    Calcul_Capital(K, tx, 1)
    Afficher(K)
Fin

' K = 1050
```

NB : Une procédure avec un seul paramètre en sortie peut aussi s'écrire comme une fonction. Normalement, tous les paramètres d'une fonction sont en entrée.

- 8) Écrire une procédure permettant de calculer la moyenne et la variance (paramètres en sortie) d'un échantillon de nombres réels stockés dans un tableau de taille N (paramètres en entrée). On supposera qu'un type Tab (tableau de réels) a été défini au préalable. Donner un exemple d'appel à cette procédure.

```
Procédure Moy_Var(t : Tab, n : entier, var m, v : réel)

Var i : entier
    somme, somme2 : réel

Début
    somme:=0
    somme2:=0
    Pour i de 1 à n faire
        somme:=somme+t(i)
        somme2:=somme2+t(i)^2
    Fin pour
    m:=somme/n
```

```

v:=sqr((N*somme2-somme^2)/n^2)
Fin
' Exemple d'appel à la procédure Moy_Var

Algorithme Stats

Const N=100

Type Tab=Tableau(1..N) de réels

Var i : entier
    don : Tab
    moy, vari : réel

Début
    Pour i de 1 à N faire
        Lire(don(i))
    Fin pour
    Moy_Var(don, N, moy, vari)
    Afficher(moy, vari)
Fin

```

- 9) Écrire une procédure permettant de trier un tableau de nombres réels positifs de taille N (paramètre en entrée/sortie). On supposera qu'un type Tab (tableau de réels) a été défini au préalable.

Méthode : Trier les valeurs contenues dans le tableau t1 dans un second tableau t2, de la plus grande à la plus petite. Pour toutes les cases du tableau t2, trouver la plus grande des valeurs contenue dans t1 (en parcourant toutes les cases de t1) et l'affecter à la case courante du tableau t2. La valeur maximum trouvée dans t1 doit être mise à -1 pour ne pas être sélectionnée à nouveau. Finalement, recopier les valeurs de t2 dans t1.

NB : Cette méthode de tri est la **moins** efficace possible. Pouvez-vous faire mieux ?

```

Procédure Tri_lent(var t1 : Tab, n : entier)

Var t2 : Tab
    i, j, jmax : entier
    max : réel

Début
    ' Traitement pour tous les éléments de t2
    Pour i:=1 à n faire
        ' Recherche du plus grand élément de t1
        ' Sauvegarde du plus grand dans la variable max
        ' et de sa position dans la variable jmax
        max:=t1(1)
        jmax:=1
        Pour j:=2 à n faire          ' Parcours de t1
            Si t1(j)>max alors
                max:=t1(j)
                jmax:=j
            Fin si
        Fin pour
        ' Affectation dans t2
        t2(i):=max
        ' Élimination du plus grand dans t1
        t1(jmax):=-1
    Fin pour
    ' On recopie t2 dans t1 pour que t1 soit trié en sortie

```

```

    Pour i:=1 à n faire
        t1(i):=t2(i)
    Fin pour
Fin

```