

Série N° 5 : Les Fichiers

Exercice 1 :

Soit le fichier NOMBRES.BIN qui contient une liste de nombres entiers. Écrire un algorithme qui affiche les nombres du fichier, leur somme et leur moyenne.

Exercice 2 :

Soit **Fintervalle** un fichier contenant des intervalles d'entier où chaque intervalle **I** est défini par deux entiers **a** et **b** inclus dans **I** ($I=[a,b]$).

- 1- Donner la déclaration d'un intervalle **I**.
- 2- Ecrire une action paramétrée **I_{max}** qui détermine le plus long intervalle du fichier **Fintervalle**.
- 3- Soit **Fent** un fichier d'entiers. Ecrire une procédure **CreerF** permettant de créer le fichier **Fent** contenant les éléments du plus long intervalle du fichier **Fintervalle** dans un ordre décroissant.
- 4- Ecrire une procédure **Supprime** qui supprime les bornes **a** et **b** de l'intervalle **I** du fichier **Fent**.

Exercice 3 (Rattrapage 2019):

Soit **FCAR** un fichier de caractères contenant une suite de mots séparés par un ou plusieurs blancs (espaces).

- 1- Ecrire une action paramétrée permettant de créer un fichier de mots **FMOT** contenant tous les mots du fichier **FCAR** sans les espaces.
- 2- Soit **C** un caractère donné. Ecrire une action paramétrée permettant de créer un fichier de mots **FPR** contenant tous les mots de **FMOT** commençant avec le caractère **C**.
- 3- Ecrire une action paramétrée permettant de créer, à partir du fichier **FMOT**, un fichier **FSTAT** contenant, pour chaque mot de **FMOT**, le mot lui-même et son nombre de répétition.

Exercice 4 :

Considérons les types d'enregistrements suivants :

Type TPoint = **Enregistrement** x,y : entier ; **Fin**;

TSpace= **Enregistrement** Poistion : Tpoint ; Superficie, Altitude :entier ; **Fin** ;

TVille = **Enregistrement**

InfoG : TSpace;

Nom :chaîne[25] ;

NBH :entier ; //nombre d'habitants

Fin;

- 1- Ecrire une fonction **Densite** qui renvoie la densité de population d'une Ville donnée.
- 2- Soit **FVille** un fichier de villes d'un pays. Ecrire un algorithme permettant de :
 - a- Remplir le fichier **FVille**.
 - b- Afficher la capitale du pays sachant que c'est la ville la plus dense.
 - c- Créer un fichier **G** contenant les villes se trouvant sur le même axe (Ox) que la capitale.
 - d- Comment peut-on trouver les villes voisines de la capitale suivant l'axe (Ox) ?

Exercice 5 (Examen 2014):

- 1) Soient F1 et F2 deux fichiers d'entiers strictement positifs et sans répétition. Ecrire un algorithme qui construit (crée) un fichier G d'entiers tel que G contient pour chaque valeur de F1 la valeur et tous ses multiples appartenant à F2 (F1 et F2 sont supposés existants).

Exemple :

F1 : 3 10 20 17
 F2 : 3 6 19 60 40 30
 G : 3 3 6 60 30 10 60 40 30 20 60 40 17

2) Ecrire un algorithme qui permet à partir du fichier résultat (G) de générer un autre fichier (H) contenant toutes les valeurs du fichier (G) (sans répétition) avec leur nombre.

Exemple :

H : 3 2 6 1 60 3 30 2 10 1 40 2 20 1 17 1

Exercice 6 (Rattrapage 2014):

Soit F un fichier d'entiers représentant des séquences de nombres séparées par un ou plusieurs zéro. Ecrire un algorithme qui réalise les traitements suivants :

- 1) A partir de F (fichier existant), crée un fichier G contenant pour chaque séquence, la moyenne des nombres qui la constituent.
- 2) Puis, Supprimer les valeurs nulles du fichier G.

Exemple :

F : 0 0 1 4 3 7 0 0 0 6 -9 2 7 -6 0 -10 3 0 0

G : 3,75 0,00 -3,50

G : 3,75 -3,50

Exercice 7 :

Soient F1 et F2 deux fichiers de chaînes de caractères. Chaque chaîne représente un mot. Ecrire un algorithme qui construit un fichier F3, tel que F3 contient les mots de F1 qui n'existent pas dans F2.

Exercices complémentaires

Exercice 8 :

- 1) Ecrire une procédure qui supprime le dernier élément du fichier F de nombre entiers.
- 2) En utilisant la procédure précédente, écrire un algorithme pour vider un fichier F existant de nombre entiers et de nom physique 'ESSAI.DAT', élément par élément. Cet algorithme devra afficher, après la suppression de chaque élément, la moyenne des éléments restants de F.

Exercice 9 :

Soit le type suivant :

```

Type   Produit = Enregistrement
                        Code : Entier ;
                        Désignation : Chaîne [ 80 ] ;
                        Prix : Réel ;
                        Fin ;
  
```

Soit F un fichier de produits. Ecrire une fonction qui vérifie si les éléments de F sont triés par ordre croissant de leur Code.

Exercice 10 :

L'utilisation des téléphones portables permet de stocker le répertoire des contacts dans deux fichiers :

- Un fichier 'TEL.DAT', enregistré sur la mémoire du téléphone ;
- Un fichier 'SIM.DAT', enregistré sur la mémoire de la carte SIM.

Chaque fichier contient des références d'un contact regroupant : un nom, un prénom et un numéro de téléphone.

Les éléments des deux fichiers sont supposés déjà triés selon le numéro de téléphone.

- 1) Donnez la syntaxe (les instructions) d'assignation et d'ouverture des deux fichiers.
- 2) Ecrire une procédure qui permet de stocker les doublons dans un autre fichier. Un élément est un doublon s'il existe (figure) à la fois dans les deux fichiers.

Exercice 11:

Soit F un fichier d'entiers (supposé existant) composé de séquences de nombres, chaque séquence est une répétition du même nombre (non nul). Toutes les séquences sont séparées par un zéro. Et aucune séquence du même nombre ne se répète dans le fichier.

F	14	14	14	0	5	5	0	29	29	29	29	0	6	6	6
Position	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

- 1) Ecrire une action paramétrée (**Compresser**) qui crée un fichier G d'enregistrement contenant pour chaque séquence le nombre représenté ainsi que la longueur de la séquence.

G	Nombre	14	5	29	6
	Longueur	3	2	4	3

- 2) En utilisant un seul parcours du fichier G et sans reparcourir le fichier F, trouver la position dans F de la plus longue séquence.
Position = 8
- 3) Ecrire une action paramétrée (**Decompresser**) qui permet de reconstruire un fichier H (de même type que F) à partir d'un fichier de même type que G.

Exercice 12 :

Soit **Fdata** un fichier d'entiers.

- 1) Ecrire une procédure **InsertF** permettant d'insérer un entier N dans un fichier trié par ordre croissant.
- 2) En utilisant cette procédure, écrire un algorithme permettant de trier le fichier **Fdata** dans un ordre croissant.