

Travaux Dirigés 9

Fichiers structurés séquentiels

Rappel :

- **Déclaration** : les mots-clé sont soulignés
type type_enregistrement=enregistrement
 champs₁ : type_champs₁ ;
 ...
 champs_n : type_champs_n ;
var nom_fichier_ram : fichier type_enregistrement ;
- **Ouverture** : nom_fichier_ram ← ouvrirFich(nom_fichier_externe, mode) ;
on se contente de trois modes 1. LECTURE : lecture seule , 2. ÉCRITURE : écriture seule avec tête d'écriture en début de fichier et 3. AJOUT : écriture seule avec tête d'écriture en fin de fichier.
- **Fermeture** : fermerFich(nom_fichier_ram) ;
- **Prédicat de détection de fin de fichier** : finFich(nom_fichier_ram) ;
retourne vrai si la tête de lecture a atteint la fin du fichier faux sinon.
- **Lecture dans un fichier texte** : lireFich(nom_fichier_ram, *variable*) ;
ignore les séparateurs au début puis lit une suite de caractères jusqu'au premier séparateur ou la fin de fichier et met la valeur convertie dans variable.
- **Lecture d'un caractère dans un fichier texte** :
lireCarFich(nom_fichier_ram, variable_car) ;
lit un seul caractère et déplace la tête de lecture vers le caractère suivant.
- **Écriture dans un fichier texte** :
ecrireFich(nom_fichier_ram, *expression*) ;
écrit dans le fichier une suite de caractères correspondant a la conversion de l'évaluation de expression en chaîne de caractères.
- **Séparateurs** : Deux constantes prédéfinies : SDC (séparateur de champ) pour séparer les champs et FDL la constante fin de ligne pour séparer les enregistrement.

Exemple : Un algorithme qui crée un fichier texte d'enregistrements (nom + age) puis affiche son contenu.

```
Algorithme GestionPrsonnel;
Type
  tpersonne=enregistrement
           nom:chaîne(15);
           age:entier;
  fin;
var
  nomFichier: fichier tpersonne;
  nomFichExterne: chaîne(30);
  nomPers:chaîne(15);
  comp,nbEnreg,agePers:entier;

debut
  { Première partie : Créer et remplir le fichier }
  lire(nomFichExterne);
  nomFichier ← ouvrirFich(nomFichExterne,écriture);
  lire(nbEnreg);{Nombre d'enregistrements à créer}
```

```

Pour comp ← 1 à nbEnreg faire
    lire(nomPers);
    écrireFich(nomFichier,nomPers,SDC);{SDC:séparateur de champs}
    lire(agePers);
    écrireFich(nomFichier,agePers,FDL);{FDL :fin de ligne}
fpour;
fermerFich(nomFichier);

{Deuxième partie : afficher le contenu du fichier }
nomFichier ← ouvrirFich(nomFichierExterne,LECTURE);
tant que non(finFich(nomFichier)) faire
    lireFich(nomFichier,nomPers);
    écrire(nomPers);
    lireFich(nomFichier,agePers);
    écrire(agePers);
ftq;
fermerFich(nomFichier);
Fin.

```

Exercice 1

1. Créer un fichier « ETUDIANT.TXT » dont les informations sont structurées de la manière suivante:

- Numéro de matricule (entier)
- Nom (chaîne de caractères)
- Prénom (chaîne de caractères)
- Moyenne (réel)

Le nombre d'enregistrements à créer est à entrer au clavier par l'utilisateur.

Afficher les informations du fichier « ETUDIANT.TXT » ainsi que la moyenne des étudiants .

2. Ajouter un nouvel enregistrement (entré au clavier) à la fin d' « ETUDIANT.TXT » et sauver le nouveau fichier sous le nom « ETUDIANT2.TXT ».

3. Insérer un nouvel enregistrement dans « ETUDIANT.TXT » en supposant que le fichier est trié relativement à la rubrique NOM et sauver le nouveau fichier sous le nom « ETUDIANT3.TXT ».

4. Supprimer dans « ETUDIANT.TXT » tous les enregistrements dont la moyenne est inférieure à 10. Sauver le nouveau fichier sous le nom « ETUDIANT4.TXT ».

Exercice 2 (Tri interne de fichiers)

Écrire un programme qui lit 10 valeurs entières d'un fichier de 10 entiers, les place dans un tableau, tri le tableau dans l'ordre croissant et place ce tableau trié dans le même fichier à la place des 10 valeurs de départ

Exercice 3

1. Écrire un programme permettant d'enregistrer dans un fichier un nombre quelconque de valeurs entières positives fournies au clavier.

2. Écrire un programme permettant de retrouver, dans le fichier précédent, une valeur de rang donné. On fournira un message dans le cas où le rang demandé dépasserait la taille du fichier.

Exercice 4

1. Écrire une fonction qui permet de créer un fichier contenant une succession d'informations formatées fournies par l'utilisateur. Chaque information correspond à un point d'un plan défini par un nom formé d'une lettre, et par deux coordonnées entières. L'exécution de cette fonction donnerait par exemple :

Nom du fichier à créer : courbe.txt

Donner le nom (* pour finir) et les coordonnées des points :

C 12 34
F 8 121
Z 152 95
X 25 74
*

2. Écrire une fonction permettant de lister le contenu du fichier formaté créé précédemment. Prendre en compte les possibilités d'erreur de lecture ou d'ouverture.

3. Écrire une fonction permettant de produire, à la demande de l'utilisateur, le contenu de n'importe quelle ligne dont le numéro est fourni. Exemple d'exécution :

Nom du fichier à consulter : courbe.txt

Votre fichier comporte 4 points

Numéro du point cherché (0 pour finir) ? : 2

Point numéro 2 est : F 8 121

Numéro du point cherché (0 pour finir) ? : 4

Point numéro 2 est : X 25 74

Numéro du point cherché (0 pour finir) ? : 6

Point incorrect

Numéro de points cherché (0 pour finir) ? : 0

A bientôt !

----- Exercices supplémentaires -----

ETLD 2014-2015 : Une ménagère fait ses courses depuis deux supérettes : une à Alger et l'autre à Blida. Elle dispose sur papier de deux listes d'historique d'achats, une pour chaque supérette, contenant les informations : article, prix, et date d'achat en jour/mois/année, ordonnées par la date d'achat. Voici un exemple de ses deux listes :

Liste d'achat supérette d'Alger		
article	Prix en DA	date
Pain	50	25/04/2014
Tricot	900	25/04/2014
Cuisinière	35000	10/01/2015
Viande	1500	12/05/2015

...

Liste d'achat supérette de Blida		
article	Prix en DA	date
Légumes	500	25/02/2014
Chemise	1800	11/03/2014
Fruits	450	08/07/2014
Viande	2000	12/06/2015

...

Cette ménagère voudrait automatiser la gestion de ses dépenses.

1. Proposer les structures de donnée pour représenter les informations d'un achat.

2. Proposer une fonction ou procédure pour remplir un tableau de liste d'achats, puis donner les appels à cette fonction pour remplir chacun des deux tableaux listAchatAlger et listAchatBlida.
3. Proposer une fonction pour déterminer à partir des tableaux listAchatAlger et listAchatBlida le montant global des dépenses entre deux dates données.
4. Proposer une fonction ou procédure pour sauvegarder les informations des deux tableaux listAchatAlger et listAchatBlida dans deux fichiers fichAchatAlger et fichAchatBlida.
5. Proposer une fonction ou procédure pour afficher à partir des fichiers fichAchatAlger et fichAchatBlida, les mois où les dépenses dépassent 30000 DA.

Remarque : On suppose que les informations des champs sont sans espace.

ETLD 2013-2014 : Une entreprise veut automatiser la gestion des carrières de ces employés. Elle dispose sur papier d'informations ayant la forme suivante (dans ce qui suit, on ne représente qu'une partie des listes):

Liste des employés			
Nom	Prénom	Poste	Date de recrutement
Benmalki	Ali	Directeur	23/05/2000
Mahmoudi	Zaki	Agent	20/04/2010
Bousri	Sami	Mécanicien	01/06/2005
Ouafi	Aissa	Agent	04/08/1998
Dardour	Imed	Chauffeur	05/03/2001
Sidhoum	Rima	Secrétaire	12/07/2001
---	---	---	---
---	---	---	---

Liste des postes	
Poste	Salaire
Agent	20000
Directeur	55000
Mécanicien	25000
Chauffeur	22000
Secrétaire	25000
---	---
---	---

1. Proposer une structure (enregistrement) pour stocker une ligne de Liste des employés et une autre pour stocker une ligne de Liste des postes.
2. Écrire une action paramétrée pour stocker ces informations dans deux tableaux de structures : Le premier pour Liste des employés et le deuxième pour Liste des postes.
3. En utilisant les tableaux de la question 2, écrire une action paramétrée qui détermine le salaire d'un employé donné.
Afin de préserver ces informations, cette entreprise décide d'utiliser des fichiers au lieu de tableaux.
4. Écrire une action paramétrée pour stocker ces informations dans deux fichiers : Le premier pour Liste des employés et le deuxième pour Liste des postes.
5. En utilisant ces deux fichiers uniquement, écrire une action paramétrée qui affiche les employés ayant un salaire inférieur strictement à 25000 DA.