

Correction de la série d'exercice N 2 (Les fichiers)

Exercice 2 :

Explication

Soit un fichier "F.dat" contenant des nombres entiers, donc on a besoin de déclarer un fichier F de type entier. F : fichier d'entier ; // File of integer

Ici le nom de fichier est donné par l'exercice "F.dat" on n'a pas besoin de demander le nom physique de l'utilisateur.

Dans cet exercice, on propose que le nom logique soit F pareil que le nom physique "F.dat".

Q1 : Avant d'écrire la procédure qui calcule la moyenne il faut écrire :

- la procédure d'initialisation qui permet de créer notre fichier F
- La procédure de Remplissage de fichier qui permet de demander de l'utilisateur de saisir les nombres et les mettre dans le fichier F.
- et on peut ajouter la procédure d'affichage qui affiche le contenu.
- pour calculer la moyenne il faut faire un accès séquentiel de tout le fichier (lire toutes les valeurs enregistrées, calcul la somme ainsi le nombre de valeurs dans le fichier et après diviser la somme sur le nombre.

Q4 : Trier le fichier F dans l'ordre décroissant

Puisque le contenu du fichier F est des nombres entiers, la simple méthode pour les trier est :

- Copie les valeurs de fichier F dans un tableau d'entier et calcul le nombre de valeur i (n).

F	T	n=8	N100										
30 7 4 60 3 8 5 2	i=8	<table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 10%; text-align: center;">30</td> <td style="width: 10%; text-align: center;">7</td> <td style="width: 10%; text-align: center;">4</td> <td style="width: 10%; text-align: center;">60</td> <td style="width: 10%; text-align: center;">3</td> <td style="width: 10%; text-align: center;">8</td> <td style="width: 10%; text-align: center;">5</td> <td style="width: 10%; text-align: center;">2</td> <td style="width: 20%; text-align: center;">.....</td> <td style="width: 10%;"></td> </tr> </table>	30	7	4	60	3	8	5	2		
30	7	4	60	3	8	5	2					

- Après appliquer un algorithme de trie du tableau (tri par sélection ou tri par insertion ou tri par bulles).

T	F
60 30 8 7 5 4 3 2	F

- Recopier les données tries du Tableau dans le fichier F.

Algorithme Exercice2;

Const

N : 100;

Var

F : fichier d'entier ; // File of integer

Q 1- Ecrire une procédure permettant de calculer la moyenne des valeurs des éléments du fichier

Procédure Initialisation (F : fichier d'entier)

Var

Début

```
Assign (F, 'F.dat') ; //Associer (F, 'F.dat') ; // la création et l'ouverture du fichier
Ouvrir (F, écriture) ;
Si(F=NULL) alors // vérification de l'existence de fichier
    Ecrire (' Fichier n'est pas créé') ;
Sinon
    Ecrire (' le Fichier est créé') ;
Fin_Si
Fermer (F) ; // la fermeture du fichier
```

Fin ;

Procédure Remplissage (F : fichier d'entier)

Var

i, x: entier ; // x est une variable entière intermédiaire entre utilisateur et le fichier

Rep : caractère ; //variable pour vérifier la réponse de l'utilisateur pour ajouter un autre nombre soit oui ou non

Début

```
Ouvrir (F, écriture); // Ouvrir (F, "w"); // ouvrir le fichier pour le mode écriture
Répéter
    Ecrire("donnez la valeur de nombre à ajouter")
    lire (nb) ;
    Ecrire (F, nb) ;
    Ecrire("Est-ce que vous voulez ajouter une nouvelle valeur si oui tapez O sinon N");
    lire (Rep) ;
    Jusqu'à (Rep = 'N') ;
Fermer (F) ;
```

Fin ;

Procédure Calcul_moyenne (F : fichier d'entier);

Var

x, Somme, nb_val: entier; // nb_val représente un compteur pour calculer le nombre de valeurs (nombres) dans le fichier

Moyenne : réel;

Début

```
Ouvrir (F, lecture); // Ouvrir (f, "r"); // ouvrir le fichier pour le mode lecture
Somme ← 0 ;
nb_val ← 0 ;
Tant que (not EOF(F)) // pour calculer la moyenne il faut faire un accès séquentiel de tout le fichier
| lire (F,x) ;
| Somme ← Somme + nb ;
| nb_val ← nb_val + 1 ;
fin_TQ ;

Si (nb_val ≠ 0) Alors
| Moyenne ← Somme / nb_val ;
| Ecrire("la Moyenne des valeurs dans le fichier =",Moyenne) ;
Sinon
| Ecrire("Fichier vide") ;
Fsi ;
Fermer (F) ;
```

Fin ;

//Affichage le contenu du fichier sur l'écran

Affichage_Fichier (F : Fichier d'entier) ;

Var

x : entier ;

Début

```
Ouvrir (F, lire);
Tant que ( not EOF(F)) faire
| Lire(F, x) ;
| Ecrire (x) ; // affichage de la valeur de x sur l'écran
fin_TQ ;
Fermer (F) ;
```

Fin ;

// Q 2- Ecrire une procédure permettant de créer deux nouveaux fichiers : le premier " F1.dat" contenant les valeurs impaires de " F" et le deuxième " F2.dat" contenant les valeurs paires de " F"

Procédure Paire_impair (F : fichier d'entier);

Var

F1, F2: fichier d'entiers; // F1 pour mettre les nombres paires et F2 pour les nombres impairs

x : entier ;

Début

```
Assign (F1, "F1.dat"); // création de F1 et F2
Assign (F2, "F2.dat");
```

```
Ouvrir (F, lire) ; //ouvrir F en mode lecture pour lire les valeurs
```

```
Ouvrir (F1, écriture) ; //ouvrir F1 en mode d'écriture des nombre paires
```

Dr.Youkana Imane

Correction Exercice 2 de la Série 2

```

    Ouvrir (F2, écriture);      //ouvrir F2 en mode d'écriture des nombre impaires
Tant que (not EOF(F)) faire
    Lire (F, x);
    Si(x mod 2 = 0)
        Ecrire(F1,x);
    Sinon
        Ecrire(F2,x);
    Finsi ;
FinTQ;
// Affichage sur l'écran le contenu des deux fichiers
Ecrire (' les valeurs paires F1 sont :');
Affichage_Fichier (F1);
Ecrire (' les valeurs impaires dans F2 sont :');
Affichage_Fichier (F2);
Fermer(F); Fermer(F1); Fermer(F2);
Fin.

```

// Q 3- Ecrire une proc_edure permettant de supprimer de F les valeurs nulles.

Procédure Supprimer_Zero_Fichier (F : fichier d'entier);

Var

F1: fichier d'entiers; // F1 pour mettre les nombres paires

x : entier ;

Debut

Assign (F1, 'F1.dat'); // création de F1

Ouvrir (F, lire); //ouvrir F en mode lecture pour lire les valeurs

Ouvrir (F1, écriture); //ouvrir F1 en mode d'écriture des nombre paires

// On doit parcourir tout le fichier F si la valeur de x ≠ 0 on le copie dans F1 sinon on ne le

prend pas

Tant que (not EOF(F)) faire

Lire(F, x);

Si(x ≠ 0)

Ecrire (F1,x);

Finsi ;

FinTQ;

// Maintenant le fichier F1 contient les valeurs ≠ 0 on doit le renommer F1 par F

Fermer(F);

Fermer(F1);

remove('F.dat');

rename('F1.dat','F.dat');

// Affichage sur l'écran le fichier F qui est maintenant contient les nombres ≠ 0

Ecrire (' les valeurs de F sont :');

Affichage_Fichier (F);

Fin.

//Q 4- Une procédure permettant de créer un fichier FTri contenant les valeurs de F triées dans l'ordre décroissant

Procédure trie_tab_selection (Var T: tableau [1..N] d'entier, n:entier); // n représente le nombre de donnée sauvegardées dans Tableau

```
Var
  i, j, Max, indMax: entier;
Début
  Pour (i ← 1 à n - 1) faire
    Max ← T[i];
    indMax ← i;
    Pour (j ← i + 1 à n) faire
      Si (Max < T[j]) alors
        indMax ← j;
        Max ← T[j];
      Fin_Si ;
    Fin_Pour ;
    t[indMax] ← T[i];
    T[i] ← Max;
  Fin_Pour ;
```

Fin ;

Procédure Tri_Fichier (F : fichier d'entier);

```
Var
  F_Tri: fichier d'entiers; // F_Tri pour créer nouveau
  x,ij : entier ;
  T: tableau [1..N] d'entier;
Début
  Ouvrir (F, lire) ; //ouvrir F en mode lecture pour lire les valeurs
  Assign (F_Tri, 'FTri.dat'); // creation du fichier F_Tri pour mettre les valeurs après le tri
  Ouvrir (F_Tri, écriture) ; //ouvrir F_Tri en mode d'écriture
  i←0 ;
  Tant que ( not EOF(F)) faire
    Lire(F, x);
    T[i] ← x;
    i← i+1 ;
  FinTQ;
  trie_tab_selection (T,i) ;
//copie les valeurs tries dans le tableau T vers le fichier F_Tri
  Pour (j ← 1 à i) faire
    Ecrire (F_Tri,t[j]);
  Fin_Pour ;
  Fermer(F); Fermer(F_Tri);

// Affichage sur l'écran le contenu du fichier F_Tri
  Ecrire ('les valeurs triées selon l'ordre décroissant du fichier F_Tri sont ');
  Affichage_Fichier(F_Tri) ;
```

Fin ;

//programme principale

Début

Initialisation (F) ;

Remplissage (F) ;

Affichage_Fichier (F) ;

Calcul_moyenne(F);

Paire_impair (F);

Supprimer_Zero_Fichier (F);

Tri_Fichier (F);

Fin.