

Solution de la série d'exercice 2 (les fichiers)

Dr. BABAHNINI Djihane

September 2020

1 Exercice 2

Soit un fichier "F.dat" contenant des nombres entiers.

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

- **Explication**

On a comme entrée un fichier dont son nom physique est F, l'extension de fichier est .dat, c'est à dire un fichier binaire, (nous avons vu dans les séances précédentes la différence entre un fichier texte et un fichier binaire

<http://elearning.univ-biskra.dz/moodle/mod/resource/view.php?id=15925>

Les données sont enregistrées dans un fichier, donc, avant de calculer la moyenne, on doit initialiser le fichier et le remplir.

Ensuite, on va calculer la moyenne, on utilisant la formule:

$$moy = \frac{\sum x}{nb} \quad (1)$$

Tel que x une valeur dans le fichier, et nb le nombre de toutes les valeurs existent dans le fichier F.

- **Algorithme**

Const

N : 100;

Procédure Init_File (FB : file of entier)

Début

Associer (FB,"F.dat") ; // création du fichier, et liaison entre le nom logique FB de fichier et le nom physique F.dat

Ouvrir (F, "w") ; //ouvrir FB en écriture

Si(FB=NULL) alors // vérification de l'existence de fichier

Ecrire ("Le fichier n'est pas créé") ;

Sinon

Ecrire ("le Fichier est bien créé") ;

Fin_Si;

Fermer (FB) ; // la fermeture du fichier

Fin ;

Procédure Remplissage (FB : file of entier)

Var

i, nb, x: entier ; // nb: le de chiffre dans le fichier

Début

Ouvrir (FB, "a"); // ouvrir le fichier en mode ajout

Ecrire("donnez le nombre de chiffres que vous voulez insérer dans le fichier");

Lire(nb);

Pour(i=1 jusqu'à nb)faire

Ecrire("donnez la valeur de nombre à ajouter")

Lire (x) ;

Ecrire (FB, x) ; //mettre la valeur x dans le fichier

```

    Fin_Pour;

    Fermer (FB);

Fin;

Procédure Calcule_moyenne (FB : file of entier)

Var

    x, S, nb: entier; // nb: nombre de valeurs dans le fichier

    Moy : réel;

Début

    Ouvrir (FB, "r"); // ouvrir le fichier en mode ajout

    S ← 0;

    nb ← 0;

    Tantque (not EOF(FB)) // accès séquentiel au fichier

        lire (FB,x) ;

        S ← S + x;

        nb ← nb + 1;

    fin_TQ ;

    Moy ← S/nb;

    Ecrire('la Moyenne des valeurs dans le fichier =',Moy) ;

    Fermer (FB) ;

Fin ;

```

2. Ecrire la 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".

- **Explication**

Nous avons vu dans les séances précédentes (semestre 1), comment séparer les valeurs paires et impaires d'un tableau T dans deux tableaux T1 et T2.

De la même façon, on va dans cet exercice séparer les valeurs paires et impaires d'un fichier binaire F, donc on va suivre les étapes suivantes:

1. ouvrir le fichier F.dat en mode lecture car on veut lire (récupérer) ses valeurs.
2. ouvrir le fichier F1.dat et F2.dat en mode ajout car on veut les remplir soit par des valeurs impaires (F1) ou par des valeurs paires (F2).
3. on accède aux valeurs de fichier F.dat par un accès séquentiel.
4. on fait un test sur la valeur récupérer x de F.dat.
5. si elle est impaire, c'est à dire le reste de sa division par deux est différent de zéro alors on la met dans le fichier F1.dat.
6. si elle est paire, c'est à dire le reste de sa division par deux est égal à zéro alors on la met dans le fichier F2.dat.

• Algorithme

Procédure Pair_impair(f: file of entier)

Var

f1, f2: file of entier;

x :entier;

Début

Associer (f,"F.dat") ;

Ouvrir (f, "o") ; //ouverture de fichier F.dat en lecture

Associer (f1,"F1.dat") ;

Ouvrir (f1, "a") ; //ouverture de fichier F1.dat en mode ajout

Associer (f2,"F2.dat") ;

Ouvrir (f2, "a") ; //ouverture de fichier F2.dat en mode ajout

Tantque (not EOF(f)) faire

 Lire(f,x); //récupération des valeurs de F.dat

 Si(x mod 2 $\neq$ 0) alors //tester la valeur récupérée

 Ecrire(f1,x); //valeur impaire, on la copie dans le fichier F1.dat

 Sinon

 Ecrire(f2,x); //valeur paire, on la copie dans le fichier F2.dat

 fin_Si ;

fin_TQ ;

```

    fermer(f);
    fermer(f1);
    fermer(f2);
Fin;

```

3. Ecrire une procédure permettant de supprimer de F les valeurs nulles.

- **Explication**

On veut dans cette exercice supprimer toutes les valeurs qui sont égaux à zéro de fichier F.dat, pour cela, on va créer un fichier intermédiaire File_Not_Null.dat, on copie dans ce fichier toutes les valeurs de F.dat différentes de zéro, à la fin de la procédure, on va supprimer le fichier F.dat, et renommer File_Not_Null.dat par F.dat

- **Algorithme**

```

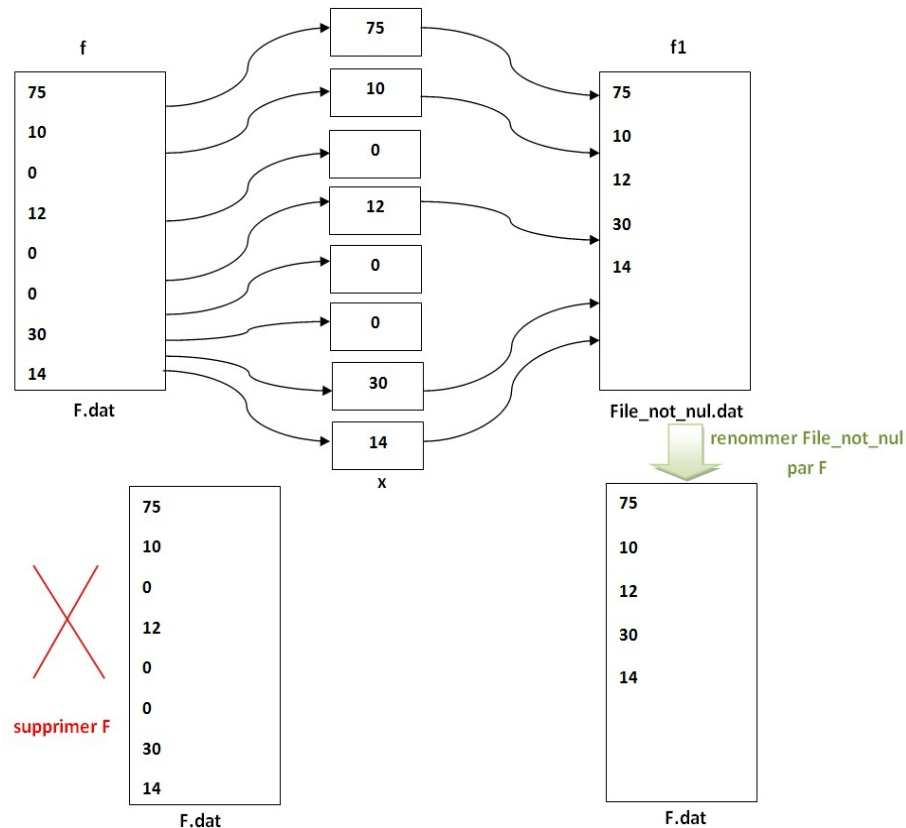
Procédure supprime_val_nul(f: file of entier)
Var
    f1: file of entier; x :entier;
Début
    Associer (f,"F.dat") ;
    Ouvrir (f, "o") ; //ouverture de fichier F.dat en lecture
    Associer (f1,"File_Not_Null.dat") ;
    Ouvrir (f1, "a") ; //ouverture de fichier F1.dat en mode ajout
    Tantque (not EOF(f)) faire
        Lire(f,x); //récupération des valeurs de F.dat
        Si(x ≠ 0) alors //tester la valeur récupérée
            Ecrire(f1,x); //valeur non nulle, on la copie dans le fichier F1.dat
        fin_Si ;
    fin_TQ ;
    fermer(f);
    fermer(f1);
    delete("F.dat");

```

```
rename("File_Not_Null.dat","F.dat");
```

```
Fin;
```

- **Tr  ce de l'algorithme**



4. Ecrire une proc  dure permettant de cr  er un fichier `FTri` contenant les valeurs de `F` tri  es dans l'ordre d  croissant.

- **Explication**

L'id  e de cet exercice est tr  s simple, on copie toutes les valeurs de fichier dans un tableau `T`, on fait le tri de `T` avec l'utilisation d'une des m  thodes de tri, par exemple le tri par s  lection, ensuite on recopie ces valeurs tri  es dans un autre fichier de sortie `F_Tri`. Donc, on a besoin de deux sous programme "`Tri_tableau`" pour faire le tri de tableau par s  lection, et "`Tri_Fichier`" pour trier les valeurs de fichier `F`.

- **Algorithme**

```
Proc  dure Tri_tableau(Var T: tableau [1..N] d'entier, n:entier)
```

```
Var
```

```

    i, j, Max, indMax: entier;
Début
    Pour (i ← 1 à n - 1) 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: file of entier)
Var
    F_Tri: file of entier; //F_Tri est le nouveau fichier trié
    x,i,j : entier ;
    T: tableau [1..N] d'entier;
Début
    Ouvrir (F, "r") ; //ouvrir F en mode lecture pour lire les valeurs
    Associer (F_Tri, "FTri.dat");
    Ouvrir (F_Tri, "a") ; //ouvrir F_Tri en mode d'ajout
    i ← 0 ;
    Tantque (not EOF(F)) faire
        Lire(F, x) ;
        T[i] ← x;
        i ← i+1 ;
    FinTQ;
    Tri_tableau (T,i) ;trier les valeurs récupérées de fichier F
    Pour (j ← 1 à i) faire

```

```
    Ecrire (F_Tri,t[j]);copie des valeurs triées dans le fichier F_tri
Fin_Pour ;
Fermer(F);
Fermer(F_Tri);
Fin;
```