

Série 1 corrigé - Algorithme (les fichiers)

Exercice 1 :

Fusionnez deux fichiers d'entiers . Le fichier résultat doit être trié .

Correction :

Procédure Fusion (E : f1 : fichier d'entiers, E : f2 : fichier d'entiers, S : f3 : fichier d'entiers)

Var : f3 : fichier d'entiers

Début

OuvrirFichier (f1, lecture)

Si EtatFichier (f1)=succès alors

OuvrirFichier (f2, lecture)

Si EtatFichier (f2)=succès alors

OuvrirFichier (f3, écriture)

Si EtatFichier (f3)=succès alors

LireFichier (f1, n1)

LireFichier (f2, n2)

Tant que (EtatFichier(f1) ? FdF ET EtatFichier(f2) ? FdF) faire

Si n1

EcrireFichier (f3, n1)

LireFichier (f1, n1)

Sinon

EcrireFichier (f3, n2)

LireFichier (f2, n2)

Si EtatFichier (f1) ? FdF alors

Répéter

EcrireFichier (f3, n1)

LireFichier(f1, n1)

Jusqu'à EtatFichier(f1)=FdF

Si EtatFichier (f2) ? FdF alors

Répéter

EcrireFichier (f3, n2)

LireFichier (f2, n2)

Jusqu'à EtatFichier (f2) = FdF

FermerFichier (f3)

Sinon écrire (« Erreur en écriture sur le fichier destination »)

FermerFichier (f2)

Sinon écrire (« Erreur de lecture sur le fichier f2 »)

FermerFichier (f1)

Sinon écrire (« Erreur en lecture sur le fichier f1 »)

Fin

EXERCICE 2 :

- 1- Écrire un algorithme qui crée le fichier MOTS.TXT contenant une série de mots (longueur maximale d'un mot: 20 caractères). La saisie des mots se terminera à l'introduction du symbole '*' qui ne sera pas écrit dans le fichier.
- 2- 2- Écrire un algorithme qui affiche le nombre de mots ainsi que la longueur moyenne des mots contenus dans le fichier MOTS.TXT.
- 3- 3- Écrire un algorithme qui crée un deuxième fichier MOTS10.TXT contenant les mots du fichier MOTS.TXT de plus de 10 caractères

Correction :

Algorithme TraiteMot;

Var F,G :Fichier de chaine[20] ;

X :chaine[20] ;

Nb :entier ; M :reel ;

Debut

/*question 1

Assigner(F,'MOTS.TXT') ; Reecrire(F) ; /*ouvrir F en écriture

Ecrire('Donner une suite de mots. Introduire le mot '*' pour arrêter la saisie') ;

Lire(X) ; /*Lire le premier mot à l'extérieur de la boucle

Tantque X≠'*'

Faire

Ecrire(F,X) ;

Lire(X) ; /*Lire le mot suivant

Fait ;

Fermer(F) ;

/*question 2

Nb←0 ; M←0 ; /*on peut utiliser M pour la somme des longueurs puis pour la moyenne

Relire(F) /*ouvrir F en lecture

Tantque Non FDF(F)

Faire

Lire(F,X) ; /*Lire un élément du fichier

M←M+Taille(X) ; Nb←Nb+1 ;

Fait ;

Si Nb≠0 Alors M←M/Nb ;

Ecrire('Nombre de mots =',Nb,' Longueur Moyenne=',M)

Sinon Ecrire('Fichier vide')

Fsi ;

Fermer(F) ;

/*question 3

Assigner(G,'MOTS10.TXT') ; Reecrire(G) ; Relire(F) /*ouvrir G en écriture et F en lecture

Tantque Non FDF(F)

Faire

Lire(F,X) ; /*Lire un élément du fichier

Si Taille(X)>10 Alors Ecrire(G,X) Fsi ;

Fait ;

Fermer(F) ; Fermer(G) ;

Fin.

EXERCICE 3 :

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 Avant suppression

G : 3,75 -3,50 Après suppression

Algorithme TraiteFile ;

Var F :Fichier de entier;

G,H :Fichier de reel ;

X,S,Nb :entier ; M :reel ;

Debut

Assigner(F,'FileF') ; Relire(F) ;

Assigner(G,'FileG') ;Reecrire(G) ;

Tantque Non FDF(F)

Faire

Lire(F,X) ;

Si X≠0

Alors S←0 ; Nb←0 ;

Tantque Non FDF(F) et X≠0

Faire S←S+X ;

Nb←Nb+1 ;

Lire(F,X) ;

Fait ;

/*traitement du dernier élément non nul du fichier, la tête est sur FDF, mais X non traité ?

Si X≠0 Alors S←S+X ; Nb←Nb+1 Fsi;

M←S/Nb ;

Ecrire(G,M) ;

Fsi ;

Fait ;

Fermer(F) ; Fermer(G) ;

/*Pour supprimer les valeurs nulles de G on utilise un fichier intermédiaire, ensuite on recopie ce

/*fichier dans G

Assigner(H,'Inter') ;Reecrire(H) ; Relire(G) ;

Tantque Non FDF(G)

Faire
 Lire(G,X) ;
 Si $X \neq 0$ Alors Ecrire(H,X) Fsi ;
 Fait ;
 Fermer(H) ; Fermer(G) ;
 /*copier H dans G
 Relire(H) ; Reecrire(G) ;
 Tantque Non FDF(H)
 Faire
 Lire(H,X) ;
 Ecrire(G,X) ;
 Fait ;
 Fermer(H) ; Fermer(G) ;
 Fin.

Exercice 4 :

Soit F1 un fichier de caractères contenant une suite de télégrammes. Chaque télégramme est constitué d'une suite de mots séparés par un ou plusieurs blancs (^). Le télégramme se termine par le mot 'FINTEL'.

Ecrire un algorithme qui permet, pour chaque télégramme, d'imprimer le texte, en respectant les conventions suivantes :

- Les mots imprimés seront séparés par un seul blanc.
- Les mots ne peuvent dépasser 12 caractères sinon ils seront tronqués à droite.
- Le texte de chaque télégramme est suivi de l'indication du nombre total de mots (tronqués ou non) et le nombre de mots tronqués.
- La fin de chaque télégramme sera indiquée par 'FINTEL'.

Correction :

Version 1 : Affichage des télégrammes sur écran

Algorithme Telegramme;
 Var F:fichier de caractère ;
 Nbmot, NbmotTq : entier;
 C:caractere;
 Mot:chaîne;
 MotTq:chaîne[12];
 Debut
 Assigner(F,'telegramme.txt') ; Relire(F) ;
 Nbmot←0; NbmotTq←0;
 Tantque Non FDF(F)
 Faire
 Lire(F,C);

Révision Bac

```
Si C<>' '  
Alors  
Mot←'' /* initialiser Mot à vide  
Tantque (Non FDF(F) et C<' ' )  
Faire  
Mot←Mot+C; /*utiliser la concaténation  
Lire(F,C);  
Fait;  
/*traiter le dernier caractère  
Si C<' ' Alors Mot←Mot+C Fsi ;  
Si mot='FINTEL'  
Alors /*traitement de la fin  
Ecrire(nbmot,' ',nbmotTq,' FINTEL') ;  
Si Non FDF(F) Alors Ecrire(' ') Fsi ;  
Nbmot←0; NbmotTq←0;  
Sinon /*traitement du mot  
Nbmot←Nbmot+1;  
Si Taille(Mot)>12  
Alors MotTq←Mot ; /* cette action permet de tronquer le mot  
NbmotTq←NbmotTq+1;  
Mot←MotTq  
Fsi ;  
Ecrire(Mot,' ' )  
Fsi  
Fsi ;  
Fait;  
Fermer(F);  
Fin.
```

Version 2 : Création d'un fichier télégrammes en respectant les contraintes

```
Dans ce Cas on aura besoin d'une Fonction pour convertir les entiers en chaine  
Algorithme Telegramme;  
Var F,FC:Fichier de caractère ;  
Nbmot, NbmotTq,I,T : entier;  
C:caractere;  
Mot:chaine;  
MotTq:chaine[12];  
Fonction IntToStr(X:entiere):chaine;  
Var R:entier; Ch:chaine;  
Debut  
Ch←'';  
Repeter  
R←X MOD 10;  
X←X DIV 10;  
Cas R Vaut
```

Révision Bac

```
0:Ch←'0'+Ch;
1:Ch←'1'+Ch;
2:Ch←'2'+Ch;
3:Ch←'3'+Ch;
4:Ch←'4'+Ch;
5:Ch←'5'+Ch;
6:Ch←'6'+Ch;
7:Ch←'7'+Ch;
8:Ch←'8'+Ch;
9:Ch←'9'+Ch;
Fincas;
Jusqu'à X=0;
IntToStr←Ch;
Fin;
Debut
Assigner(F,'telegramme.txt') ; Relire(F) ;
Assigner(FC,'telegrammeC.txt') ; Reecrire(FC) ;
Nbmot←0; NbmotTq←0;
Tantque Non FDF(F)
Faire
Lire(F,C);
Si C<>' '
Alors
Mot←'' /* initialiser Mot à vide
Tantque (Non FDF(F) et C<' ' )
Faire
Mot←Mot+C; /*utiliser la concaténation
Lire(F,C);
Fait;
/*traiter le dernier caractère
Si C<' ' Alors Mot←Mot+C Fsi ;
Si Mot='FINTEL'
Alors /*traitement de la fin
Mot←IntToStr(nbmot)+' '+IntToStr(nbmotTq)+' FINTEL' ;
Si Non FDF(F) Alors Mot←Mot+' ' Fsi ;
T←Taille(Mot) ;
Pour I←1 à T Faire Ecrire(FC,Mot[I]) Fait ;
Nbmot←0; NbmotTq←0;
Sinon /*traitement du mot
Nbmot←Nbmot+1;
Si Taille(Mot)>12
Alors MotTq←Mot ; /* cette action permet de tronquer le mot
NbmotTq←NbmotTq+1;
Mot←MotTq
Fsi ;
Mot←Mot+' ' ; /* ajouter un blanc
```

Révision Bac

T ← Taille(Mot) ;
Pour I ← 1 à T Faire Ecrire(FC, Mot[I]) Fait
Fsi
Fsi ;
Fait;
Fermer(F); Fermer(FC) ;
Fin.