

P0203-3 : Structures simples de fichiers

Problème : Équilibrage et Tassement de fichiers.

Soit un fichier TÔF défini (en Z) comme suit

F UN FICHIER DE (ENTIER , VECTEUR (10) DE CHAINES)
ENTETE (ENTIER , ENTIER)
BUFFER B1 , B2

{ Chaque bloc contient le nombre d'articles et un tableau d'articles }
{ L'entête est composé de : nombre d'articles, nombre de blocs }

1. A partir de N articles (article = clé = chaîne de caractères) générés aléatoirement, créer un fichier TÔF (Tableau non ordonné de blocs et format fixe des articles) en remplissant chaque bloc à X articles. X est généré aléatoirement et varie pour chaque bloc.
2. Écrire un module Equilibrer_partiel (I, J) qui répartit équitablement les articles figurant dans les blocs I, I+1, ..., J. On suppose qu'on dispose d'une table d'articles en mémoire pouvant contenir jusqu' à Max blocs(Max donné) . (J - I + 1) est donc inférieure ou égale à Max.
3. Utiliser Equilibrer_partiel(I, J) pour équilibrer tout le fichier (par tranche de Max blocs).
4. Le module qui suit tasse (décale) tous les articles du fichier vers les premiers blocs de la manière suivante : remplit le bloc 1 à partir des blocs suivants 2, 3, ... jusqu'à ce que le bloc 1 devient plein, remplit le bloc 2 à partir des blocs suivants jusqu'à ce que le bloc 2 devient plein, Etc. A la fin de l'opération, le nombre de blocs du fichier reste inchangé. Le module transférer(B2, B1) ajoute les articles ou une partie des articles du bloc B2 vers B1.

```
ACTION Tasser ;  
SOIT  
  I, J DES ENTIERS ;  
DEBUT  
  I := 1 ; J := 2 ;  
  LIREDIR ( F , B1 , 1 ) ;  
  TQ J <= ENTETE ( F , 2 )  
    Action 1 ;  
    APPEL Transférer ( B2 , B1 ) ;  
    SI Test  
      Action2 ; Action3 ; Action4  
    FSI ;  
    J := J + 1  
  FTQ ;  
  Action5 ;  
  POUR J := I, ENTETE ( F , 2 )  
    Action6
```

FPOUR
FIN

Compléter l'action Tasser en retrouvant les 6 actions et le test Test. En déduire le module Transférer. La mesurer. Une action est soit une affectation, soit une lecture ou une écriture.

NB. La fonction aleachaine génère une chaîne de caractères. La fonction aleanombre(n) génère un nombre entre 1 et n. La fonction Mod(a, b) retourne le reste de la division de a par b.

Corrigé P0203-3

Structures de données / 2ème année 'Ingénieur' / 2002 - 2003
Corrigé (en Z) de l'EMD n°3

SOIENT

F UN FICHER DE (ENTIER , VECTEUR (10) DE CHAINES) ENTETE
(ENTIER , ENTIER) BUFFER B1 , B2 ;
{ fichier de blocs contenant le nombre d'articles et un tableau d'articles }
{ entête : nombre d'articles, nombre de blocs }
Creer , Imprimer , Equilibrer , Tout_equilibrer , Tasser DES ACTIONS ;
Transferer , Copier DES ACTIONS ;
Table UN VECTEUR (50) DE CHAINES ;
Limite UN ENTIER ;
{ Limite : nombre maximal de blocs que l'on peut mettre dans Table }

DEBUT

Limite := 5 ;
APPEL Creer ;
APPEL Imprimer ;
Ecrire ('Equilibrage partiel') ;
APPEL Equilibrer (1 , Limite) ;
APPEL Imprimer ;
Ecrire ('Equilibrage total') ;
APPEL Tout_equilibrer ;
APPEL Imprimer ;
Ecrire ('Tassement') ;
APPEL Tasser ;
APPEL Imprimer ;

FIN

/***** Chargement de n articles avec un chargement de x % par bloc *****/

ACTION Creer ;

SOIENT

I , K , N , X , Nbblocs DES ENTIERS ;

DEBUT


```

OUVRIR ( F , 'f.pas' , 'N' ) ;
I := 0 ;
Nbblocs := 0 ;
N := 105 ;
AFF_ENTETE ( F , 1 , N ) ;
TQ I < N :
    K := 0 ;
    X := ALEANOMBRE ( 10 ) + 1 ;
    TQ ( K < X ) ET ( I < N )
        K := K + 1 ;
        I := I + 1 ;
        AFF_ELEMENT ( STRUCT ( B1 , 2 ) [ K ] , ALEACHAINE ( 10 ) )
    FTQ ;
    AFF_STRUCT ( B1 , 1 , K ) ;
    Nbblocs := Nbblocs + 1 ;
    ECRIRESEQ ( F , B1 ) ;
    FTQ ;
    AFF_ENTETE ( F , 2 , Nbblocs ) ;
    FERMER ( F ) ;
FIN

```

/***** Impression des articles du fichier *****/

ACTION Imprimer ;

SOIENT

I , K DES ENTIERS ;

DEBUT

I := 0 ;

OUVRIR (F , 'f.pas' , 'A') ;

TQ NON FINFICH (F)

I := I + 1 ;

ECRIRE ('B L O C n° ' , I) ;

LIRESEQ (F , B1) ;

POUR K := 1 , STRUCT (B1 , 1)

ECRIRE (ELEMENT (STRUCT (B1 , 2) [K]))

FPOUR

FTQ ;

FIN

/***** Equilibrage partiel *****/

ACTION Equilibrer (I , J) ;

SOIT

Ind , K , I , J , M DES ENTIERS ;

Quotient , Reste DES ENTIERS ;

DEBUT

Ind := 0 ;

POUR K := I , J

LIREDIR (F , B1 , K) ;

```

    POUR M := 1, STRUCT ( B1, 1 )
        Ind := Ind + 1 ;
        AFF_ELEMENT ( Table [ Ind ], ELEMENT ( STRUCT ( B1, 2 ) [ M ] ) )
    FPOUR ;
FPOUR ;
Quotient := Ind / ( J - I + 1 ) ;
Reste := MOD ( Ind, ( J - I + 1 ) ) ;
Ind := 0 ;
POUR M := I, J
    POUR K := 1, Quotient
        Ind := Ind + 1 ;
        AFF_ELEMENT ( STRUCT ( B1, 2 ) [ K ], ELEMENT ( Table [ Ind ] ) )
    FPOUR ;
    AFF_STRUCT ( B1, 1, Quotient ) ;
    SI Reste > 0
        Reste := Reste - 1 ;
        Ind := Ind + 1 ;
        AFF_ELEMENT ( STRUCT ( B1, 2 ) [ K ], ELEMENT ( Table
[ Ind ] ) ) ;
        AFF_STRUCT ( B1, 1, Quotient + 1 ) ;
    FSI ;
    ECRIREDIR ( F, B1, M ) ;
FPOUR ;

```

FIN

/**** Equilibrage total *****/

ACTION Tout_equilibrer ;

SOIT

N_intervalles UN ENTIER ;

Reste, I DES ENTIERS ;

DEBUT

N_intervalles := ENTETE (F, 2) / Limite ;

Reste := MOD (ENTETE (F, 2), Limite) ;

POUR I := 0, N_intervalles - 1

APPEL Equilibrer ((I * Limite) + 1, (I * Limite) + Limite)

FINPOUR ;

SI Reste <> 0

APPEL Equilibrer (N_intervalles * Limite + 1, ENTETE (F, 2))

FSI

FIN

/**** Copier Buf2 dans Buf1 *****/

ACTION Copier (Buf2, Buf1) ;

SOIT

Buf1, Buf2 DES (ENTIER, VECTEUR (10) DE CHAINES) ;

DEBUT

Buf1 := Buf2 ;

FIN

/**** Transférer les articles de Buf1 dans Buf2 *****/

ACTION Transferer (Buf1 , Buf2)

SOIT

Buf1 , Buf2 DES (ENTIER , VECTEUR (10) DE CHAINES) ;

I , J DES ENTIERS ;

DEBUT

POUR I := 1 , MIN (STRUCT (Buf1 , 1) , 10 - STRUCT (Buf2 , 1))

AFF_ELEMENT (STRUCT (Buf2 , 2) [STRUCT (Buf2 , 1) + 1] ,

ELEMENT (STRUCT (Buf1 , 2) [STRUCT (Buf1 , 1)])) ;

AFF_STRUCT (Buf1 , 1 , STRUCT (Buf1 , 1) - 1) ;

AFF_STRUCT (Buf2 , 1 , STRUCT (Buf2 , 1) + 1) ;

FPOUR

FIN

/**** Tasser *****/

ACTION Tasser ;

SOIT

I , J DES ENTIERS ;

DEBUT

I := 1 ;

J := 2 ;

LIREDIR (F , B1 , 1) ;

TQ J <= ENTETE (F , 2)

LIREDIR (F , B2 , J) ;

APPEL Transferer (B2 , B1) ;

SI (STRUCT (B1 , 1) = 10) OU (J = ENTETE (F , 2))

ECRIREDIR (F , B1 , I) ;

I := I + 1 ;

APPEL Copier (B2 , B1)

FSI ;

J := J + 1

FTQ ;

AFF_STRUCT (B2 , 1 , 0) ;

POUR J := I , ENTETE (F , 2)

ECRIREDIR (F , B2 , J)

FPOUR

FIN