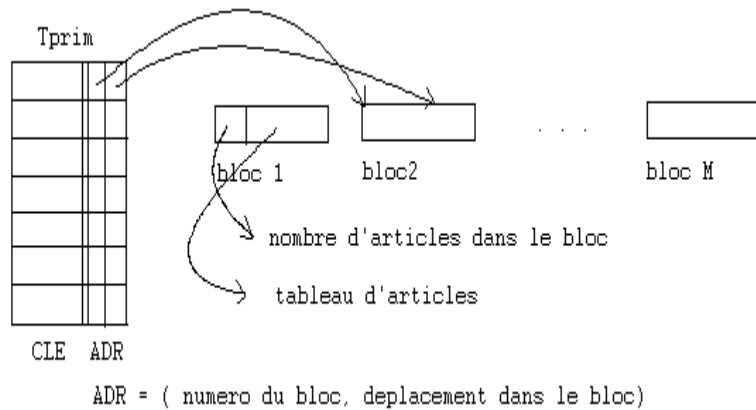


Solution TD index

Exercice 2 :

Un exemple d'organisation de fichier avec index primaire est le suivant :



Tprim étant la table d'index.

Caractéristiques

N : nombre courant d'éléments dans *Tprim*.

Dernierbloc : numéro du dernier bloc du fichier.

Deplacement : déplacement dans

le bloc.

Max_bloc : nombre maximum de blocs du fichier.

Description

```
TYPE T = STRUCTURE
```

```
  Clé : Typeclé
```

```
  Adr : Typeadresse
```

```
FIN
```

```
TYPE Typedresse = STRUCTURE
```

```
  Numbloc : ENTIER
```

```
  Depl : ENTIER
```

```
FIN
```

```
VAR Tprim : TABLEAU[1..M.] DE T
```

Définition du bloc du fichier

```
TYPE Typebloc = STRUCTURE
```

```
  Nb : ENTIER
```

```
  Tab : TABLEAU[1..B] DE Typearticle
```

```
FIN
```

```
TYPE Typearticle = STRUCTURE
```

```
  Clé : Typeclé
```

```
  Info : Typeqq
```

```
FIN
```

Algorithme de recherche/insertion

Soit F le fichier de données et Bloc une zone mémoire de type Typebloc.

```
{ Recherche dichotomique sur l'index primaire }
Trouv := FAUX
Bi := 1
Bs := N { Nombre d'éléments dans Tprim }
TANTQUE Bi <= Bs ET NON Trouv :
    Milieu := (Bi+Bs) DIV 2
    SI Tprim(Milieu).Clé = Clé
        Trouv := VRAI
    SINON
        SI Clé < Tprim(Milieu).Clé
            Bs := Milieu - 1
        SINON
            Bi := Milieu + 1
    FSI
FSI
FINTANTQUE
SI Trouv : " Clé existe "
SINON
    { Bi pointe la position où l'élément devrait être inséré }
    SI N = M : " Saturation de la table d'index "
    SINON
        Possible := VRAI
        { Insertion de l'article dans le fichier }
        SI Déplacement < B:
            Lirebloc(F,Dernierbloc, Bloc)
            Bloc.Nb := Bloc.Nb + 1
            Bloc.Tab(Bloc.Nb) := Article
            Déplacement := Bloc.Nb
            Ecrirebloc(F, Dernierbloc, Bloc)
        SINON
            { Le dernier bloc est plein }
            Dernierbloc := Dernierbloc + 1
            SI Dernierbloc <= Max_bloc :
                Bloc.Nb := 1
                Bloc.Tab(1) := Article
                Déplacement := 1
                Ecrirebloc(F, Dernierbloc, Bloc)
            SINON Possible := FAUX
                " Fichier saturé "
        FSI
    FSI
    { Insertion ( Clé, (Dernierbloc, Deplacement)) dans la table d'index Tprim }
    SI Possible :
        POUR I = N, Bi, - 1 :
```

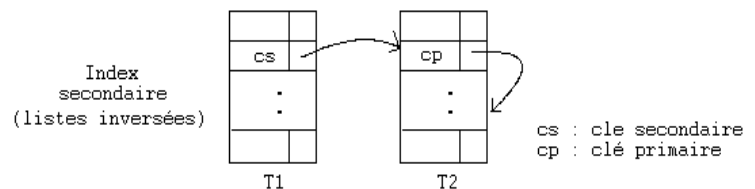
```

    Tprim(I+1) := Tprim(I)
  FINPOUR
  Tprim(Bi).Clé := Clé ;   N := N + 1
FSI
FSI
FSI

```

Exercice 3 :

Schéma



Description

```

TYPE Type1 = STRUCTURE
  Clésec : Typeclé
  Têtelist : ENTIER
FIN

```

```

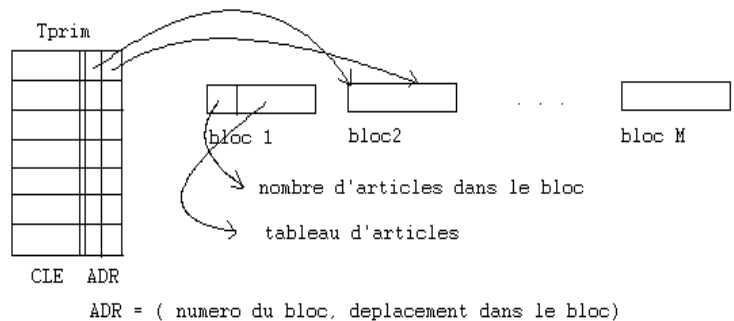
TYPE Type2 = STRUCTURE
  Cléprim : Typeclé
  Suiv : ENTIER
FIN

```

```

VAR T1 : TABLEAU[1..N1] DE Type1
VAR T2 : TABLEAU[1..N2] DE Type2

```



T1 et T2 désignent l'index secondaire sous forme de listes inversées.

```

TYPE Type3 = STRUCTURE
  Cléprim : Typeclé
  Adr : Typeadresse
FIN

```

```

TYPE Typedresse = STRUCTURE
  Numbloc : ENTIER
  Depl : ENTIER
FIN

```

```

VAR Tprim : TABLEAU[1..N.] DE Type3

```

Définition du bloc du fichier

```

TYPE Typebloc = STRUCTURE
  Nb : ENTIER
  Tab : TABLEAU[1..B] DE Typearticle
FIN

```

```

TYPE Typearticle = STRUCTURE
  Clé : Typeclé
  Info : Typeqq
FIN

```

Caractéristiques

N : nombre courant d'éléments dans Tprim.
 Dernierbloc : numéro du dernier bloc du fichier.
 Deplacement : déplacement dans le bloc.
 Max_bloc : nombre maximum de blocs du fichier.
 Nc1 : nombre courant d'éléments dans T1.
 Nc2 : nombre courant d'éléments dans T2.

Algorithme de recherche/insertion

On commence par dérouler exactement le même algorithme que précédemment. Si la clé primaire n'est pas trouvée, elle est insérée dans le fichier (l'article est réduit à sa clé) et dans la table d'index primaire. Si le fichier dispose en plus d'un index secondaire, on déroule ce qui suit:

```

{Recherche dichotomique de Clésec dans T1}
  Trouv := FAUX
  Bi := 1
  Bs := Nc1 { Nombre d'éléments dans T1 }
  TANTQUE Bi <= Bs ET NON Trouv :
    Milieu := (Bi+Bs) DIV / 2
    SI T1(Milieu).Clésec = Clésec
      Trouv := VRAI
    SINON
      SI Clé < T1(Milieu).Clésec
        Bs := Milieu - 1
      SINON
        Bi := Milieu + 1
    FSI
  FSI
FINTANTQUE
Possible := VRAI
SI NON Trouv :
  {Bi pointe la position où devrait être inséré l'élément}
  SI Nc1 = N1 : " Saturation de T1" ; Possible := FAUX

  SINON
    POUR I = Nc1, Bi, - 1 :
      T1(I+1) := T1(I)
    FINPOUR
    T1(Bi).Clésec := Clésec
    T1(Bi).Têtelist := NIL
    Nc1 := Nc1 + 1
  FSI
  SI Possible :

```

```

Allouer(R)
SI R # 0 :
    {Insertion au début}
    Aff_Val(R, Cleprim)
    Aff_Adr(R, T1(Bi).Têtelist )
    T1(Bi).Têtelist := R
SINON " Saturation de T2" FSI
FSI
SINON "Impossible" FSI

```

Les modules Allouer et Aff_Adr et Aff_Val sont définis comme suit :

Module Allouer(R) : délivre un emplacement libre R dans la table T2 s'il existe.

```

Nc2 := Nc2 + 1
SI Nc2 <= N2 :
    R <-- Nc2
SINON
    R := 0
FSI

```

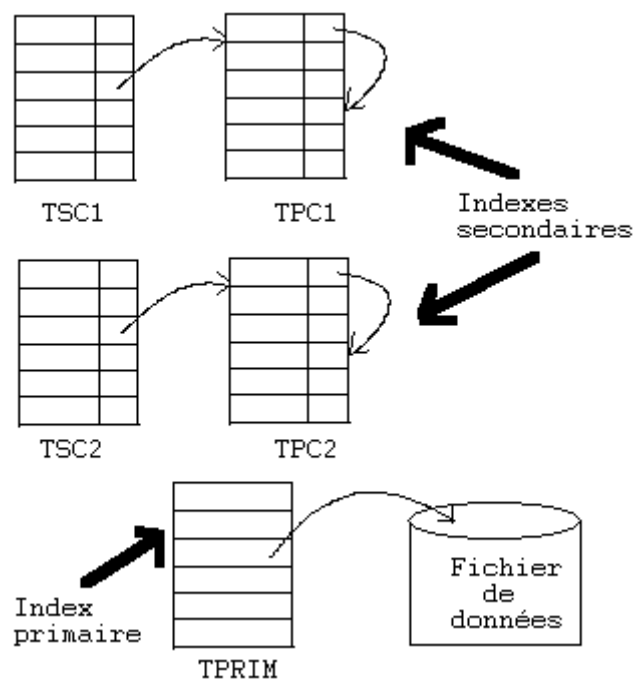
Notons qu'une suppression dans un tel fichier ne touche pas l'index secondaire.

Module Aff_Adr(L, K) : dans le champ Suiv de l'entrée L de T2, met l'indice K, c'est à dire :
 $T2(L).Suiv := K$

Module Aff_Val(L, V) : dans le champ Cléprim de l'entrée L de T2, met la clé V, c'est à dire :

$T2(L).Cléprim := V$

Exercice 4:



Description algorithmique

Index secondaires :

TYPE T1 = STRUCTURE

Clésec : Typeclé

Têtelist: ENTIER

FIN

TYPE T2 = STRUCTURE

Cléprim : Typeclé

Suiv: ENTIER

FIN

VAR Tsc1, Tsc2 : TABLEAU[1..M] DE T1

VAR Tpc1, Tpc2 : TABLEAU[1..M] DE T2

Index primaire :

TYPE T = STRUCTURE

Cléprim : Typeclé

Adr : Typeadresse

FIN

VAR Tprim : TABLEAU[1..P] DE T

Recherche de tous les articles de clés C1 et C2 données

Recherche dichotomique de C1 sur TSC1 --> Trouv, I

SI NON Trouv :

" aucun article "

SINON

Recherche dichotomique de C2 sur TSC2 --> Trouv', I'

SI NON Trouv' :

" aucun"

SINON

a) Former l'ensemble I, intersection des listes I et I' contenues dans TPC1 et TPC2.

b) Pour chaque élément de I, faire une recherche dichotomique dans TPRIM pour récupérer l'article dans le fichier de données.

Dans la pratique a) et b) se déroulent en parallèle.

Suppression d'article

Une suppression d'article se fait uniquement au niveau de la table d'index primaire.

Nouvelle structure de TPRIM

TYPE T = STRUCTURE

Cléprim : Typeclé

Effacé : Booléen

Adr : Typeadresse

FIN

VAR Tprim : TABLEAU[1..P] DE T

Algorithme de suppression

```
Trouv := FAUX
Bi := 1
Bs := N { Nombre d'éléments dans Tprim }
TANTQUE Bi <= Bs ET NON Trouv :
    Milieu := Ent(Bi+Bs) / 2 )
    SI Tprim(Milieu) = Clé
        Trouv := VRAI
    SINON
        SI Clé < Tprim(Milieu)
            Bs := Milieu - 1
        SINON
            Bi := Milieu + 1
    FSI
FSI
FINTANTQUE
SI Trouv :
    SI Tprim(Milieu).Effacé :
        " Article inexistant "
    SINON
        " Tprim(Milieu).Effacé := VRAI FSI
SINON " Article inexistant " FSI
```