

Pour chaque partie, veuillez répondre sur une feuille d'examen à part

Partie A (8 points)

(Répondre sur une feuille d'examen à part)

Questions de cours sur 5 pts

- Donner les structures de tables en RAM pour les méthodes:
 - Index primaire à un niveau
 - Index primaire à 2 niveaux
 - Index secondaire sous forme de listes inversées
- Dire comment se fait une suppression d'article de clé primaire donnée dans la méthode d'accès multicritères ?
- Définir la méthode « Listes inversées » ?

Exercice 1 sur 3 pts

(a) - Que fait cet algorithme ? Quelle est la structure du fichier utilisée ?

Remarque importante : Les blocs i et $i+1$ sont pleins et le bloc $i+2$ n'est pas plein

```
// Un fichier est défini comme suit :
F un FICHER de TBLOC BUFFER Buf, Buf2 ENTETE (entier) ;
// L'entête(F,1) désigne le nombre de blocs dans le fichier
Type TBLOC = Structure
  nb : entier ; //Nombre d'articles dans bloc
  Tab : tableau(1..B) caractère ; // Tableau contenant au maximum B articles de chaînes de caractères
Fin
...
// Dans Buf on a déjà lu le bloc i+2
// Quotient(a,b) : entier c'est une fonction qui calcule le quotient de a sur b et retourne un entier
// Reste(a,b) : entier c'est une fonction qui calcule le reste de a sur b et retourne un entier
Q := QUOTIENT(2*B+Buf.nb , 3);
R := RESTE(2*B+Buf.nb , 3);
// Mise à jour bloc I+2
// Le bloc I+2 contiendra Q+R articles
// Décalage dans le bloc I+ 2
j := 0;
  Pour k := Buf.nb jusqu'à 1 pas -1
    Buf.Tab[Q+R-j] := Buf.Tab[k];
  j ++;
FinPour
// Prendre ((B-Q)*2) articles du bloc I+1
LIREDIR (F , Buf2 , I+1);
j := 1 ;
  Pour k := (2*Q - B + 1) jusqu'à B
    Buf.Tab[j] := Buf2.Tab[k] ;
  j ++;
FinPour
Buf.nb := Q+R;
ECRIREDIR (F , Buf , I+2);
LIREDIR (F , Buf , I);
```



```

// Mise à jour bloc I+1
// Décalage dans le bloc I+ 1
j := 0 ;
  Pour k := 2*Q - B jusqu'à 1 pas -1
    Buf2.Tab[Q-j] := Buf2.Tab[k];
    j ++ ;
  FinPour
// Prendre (B-Q) articles du bloc I
j := 1 ;
  Pour k := Q+1 jusqu'à B
    Buf2.Tab[j] := Buf.Tab[k];
    j ++ ;
  FinPour
Buf2.nb := Q;
ECRIREDIR (F , Buf2 , I+1);
// Mise à jour bloc I
Buf.nb := Q;
ECRIREDIR (F , Buf , I);
...

```

(b) – Soit l'exemple suivant représentant 3 blocs consécutifs I, I+1 et I+2 d'un fichier donné.

En exécutant le programme ci-dessus, représentez les contenus de chaque bloc I, I+1 et I+2 tels que B=8 et les blocs I et I+1 sont pleins et le bloc I+2 n'est pas plein.

Avant exécution :

	8								
Bloc I	a	b	c	d	e	f	g	h	
	8								
Bloc I+1	i	j	k	k	m	n	o	p	
	3								
Bloc I+2	q	r	s						

Après exécution : ???

Bloc I									
Bloc I+1									
Bloc I+2									

Partie B (8 points)

(Répondre sur une feuille d'examen à part)

Exercice 2 Hachage Linéaire (dynamique).

But de l'exercice : Ecriture de la procédure de **recherche d'une valeur**, et la procédure d'**insertion d'une valeur**.

Les données :

- 2 fonctions de hachage h_1 et h_{i+1}
- 2 fichiers :
 - Fp : fichier zone primaire du type T \overline{O} F, ses enregistrements sont réduits à un entier donc on peut dire que c'est un fichier de valeur et non un fichier d'enregistrement.
 - Fd : fichier zone de débordement associé à Fp du type L \overline{O} F. Si insertion en zone de débordement alors elle sera faite en tête de liste si il y a de la place sinon une nouvelle tête est allouée.
- Les tailles des blocs de Fp et Fd ne sont pas forcément égales.

Travail demandé :

1. En plus des deux procédures citées plus haut, donnez les déclarations complètes du programme principal.

Liste non exhaustive de type et variables à utiliser :

- Les types : Tblocp, Tblocd, Fp (fichier primaire), Fd (fichier débordement), entetep, enteted
- Les variables : bufp, bufd // si vous avez besoin d'autre buffer alors rajouter un num ex : bufd2.

2. Que faut il faire pour accélérer la recherche en zone primaire et quelles sont les conséquences sur les autres procédures (rech, ins, supp).

Fonctions fournies :

Le modèle des fichiers, condition d'éclatement, Eclatement d'un bloc et sa zone de débordement.

Remarque : La suppression qui n'est pas demandée est physique.

Partie C (4 points)

(Répondre sur une feuille d'examen à part)

Exercice 3

Nous souhaitons implémenter efficacement un modèle de file d'attente (FIFO) en MS en utilisant des fichiers sous forme de listes de blocs. Les éléments de la file sont des enregistrements d'un type 'T_enreg' donné ayant un format fixe.

Nous disposons de deux buffers en MC.

- Donnez une **solution efficace** pour l'implémentation du modèle, permettant la réutilisation des blocs libérés.

Rappel : Le modèle des files d'attente FIFO est constitué des opérations suivantes :

{ CreerFile(F):initialise F , FileVide(F):teste si F est vide , Enfiler(F, e):rajoute e en queue de file , Defiler(F, e):supprime et récupère dans e l'élément en tête de file }