

Chaque exercice doit être traité sur une double feuille à part.

1) Intersection de 2 fichiers

Soient F1 et F2 deux fichiers volumineux contenant le même type d'enregistrement. On voudrait mettre dans un nouveau fichier F3, tous les enregistrements qui existent en même temps dans F1 et dans F2 (l'intersection de F1 et F2). On considère les hypothèses suivantes :

- Les clés sont uniques, donc si la clé d'un enregistrement e1 de F1 est égale à la clé d'un enregistrement e2 de F2, alors $e1 = e2$
- Les tailles en nombre de blocs ($N1$ et $N2$) des deux fichiers F1 et F2 sont grandes et à peu près égale. C'est à dire $N1 \approx N2$
- La capacité maximale (b) des blocs est fixe pour tous les fichiers, de plus le facteur de chargement de tous les fichiers, est au voisinage de 100%
- Les enregistrements sont à taille fixe, le fichier F1 est ordonné et le fichier F2 ne l'est pas
- Tous les fichiers sont vus comme 'Tableaux'
- Il n'y a pas d'enregistrement supprimé logiquement.

a) Donnez un algorithme permettant de **réaliser l'intersection des fichiers F1 et F2** dans un nouveau fichier F3. Donc F3 ne contiendra que les enregistrements qui existent à la fois dans F1 et dans F2.

b) Donnez le coût de cette opération en pire cas et donnez les déclarations en langage C pour pouvoir manipuler ce type de fichier.

2) Compactage d'un fichier

Soit F un fichier TOF (vu comme tableau, ordonné et les enregistrements sont à taille fixe). La capacité maximale d'un bloc est fixée à b enregistrements. On désire **réduire la taille du fichier** en récupérant les espaces occupés par les enregistrements supprimés logiquement ainsi que les vides pouvant exister dans chaque blocs. Donc après cette opération, le fichier ne contiendra plus d'enregistrement supprimé logiquement et tous les blocs à part peut être le dernier seront remplis à 100% (le fichier restera ordonné).

Les caractéristiques du fichiers sont :

- Le nombre de blocs utilisés (NbLc)
- Le nombre d'enregistrements insérés (nbIns)
- Le nombre d'enregistrements supprimés logiquement (nbSup)

a) Donnez un algorithme qui **réalise cette opération en une seule passe**, sur le même fichier (sans créer un nouveau fichier ou de nouveau blocs) et en utilisant en mémoire centrale pas plus de 2 buffers.

b) Donnez le coût de cette opération en pire cas et en moyenne.

F3 = Intersection de F1 et F2

F1

10,aa,bb	<input type="checkbox"/>
20,cc,dd	
30,ee,ff	

40,gg,hh	<input type="checkbox"/>
50,aa,ii	
60,jj,aa	

70,cc,bb	<input type="checkbox"/>
80,zz,ee	
90,tt,uu	

100,ll,ss	<input type="checkbox"/>
110,aa,rr	
120,gg,ff	

130,yy,vv	<input type="checkbox"/>
150,kk,zz	

F2

105,xx,yy	<input type="checkbox"/>
200,rr,rr	
70,cc,bb	

18,tt,hh	<input type="checkbox"/>
50,aa,ii	
130,yy,vv	

25,hh,nn	<input type="checkbox"/>
80,zz,ee	
10,aa,bb	

35,zz,xx	<input type="checkbox"/>
87,dd,bb	
55,ww,xx	

190,ff,qq	<input type="checkbox"/>
30,ee,ff	
150,kk,zz	

F3

70,cc,bb	<input type="checkbox"/>
50,aa,ii	
130,yy,vv	

80,zz,ee	<input type="checkbox"/>
10,aa,bb	
30,ee,ff	

150,kk,zz	<input type="checkbox"/>

Compactage d'un fichier
* : enregistrement supprimé logiquement

Avant

<table><tr><td>10,aa,bb</td><td><input type="checkbox"/></td></tr><tr><td>20,cc,dd</td><td></td></tr><tr><td></td><td></td></tr></table>	10,aa,bb	<input type="checkbox"/>	20,cc,dd				<table><tr><td>40,gg,hh,*</td><td><input type="checkbox"/></td></tr><tr><td>50,aa,ii</td><td></td></tr><tr><td></td><td></td></tr></table>	40,gg,hh,*	<input type="checkbox"/>	50,aa,ii				<table><tr><td>70,cc,bb</td><td><input type="checkbox"/></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr></table>	70,cc,bb	<input type="checkbox"/>					<table><tr><td>100,ll,ss</td><td><input type="checkbox"/></td></tr><tr><td>110,aa,rr,*</td><td></td></tr><tr><td>120,gg,ff</td><td></td></tr></table>	100,ll,ss	<input type="checkbox"/>	110,aa,rr,*		120,gg,ff		<table><tr><td>130,yy,vv</td><td><input type="checkbox"/></td></tr><tr><td>150,kk,zz</td><td></td></tr><tr><td></td><td></td></tr></table>	130,yy,vv	<input type="checkbox"/>	150,kk,zz			
10,aa,bb	<input type="checkbox"/>																																	
20,cc,dd																																		
40,gg,hh,*	<input type="checkbox"/>																																	
50,aa,ii																																		
70,cc,bb	<input type="checkbox"/>																																	
100,ll,ss	<input type="checkbox"/>																																	
110,aa,rr,*																																		
120,gg,ff																																		
130,yy,vv	<input type="checkbox"/>																																	
150,kk,zz																																		

Après

<table><tr><td>10,aa,bb</td><td><input type="checkbox"/></td></tr><tr><td>20,cc,dd</td><td></td></tr><tr><td>50,aa,ii</td><td></td></tr></table>	10,aa,bb	<input type="checkbox"/>	20,cc,dd		50,aa,ii		<table><tr><td>70,cc,bb</td><td><input type="checkbox"/></td></tr><tr><td>100,ll,ss</td><td></td></tr><tr><td>120,gg,ff</td><td></td></tr></table>	70,cc,bb	<input type="checkbox"/>	100,ll,ss		120,gg,ff		<table><tr><td>130,yy,vv</td><td><input type="checkbox"/></td></tr><tr><td>150,kk,zz</td><td></td></tr><tr><td></td><td></td></tr></table>	130,yy,vv	<input type="checkbox"/>	150,kk,zz			
10,aa,bb	<input type="checkbox"/>																			
20,cc,dd																				
50,aa,ii																				
70,cc,bb	<input type="checkbox"/>																			
100,ll,ss																				
120,gg,ff																				
130,yy,vv	<input type="checkbox"/>																			
150,kk,zz																				