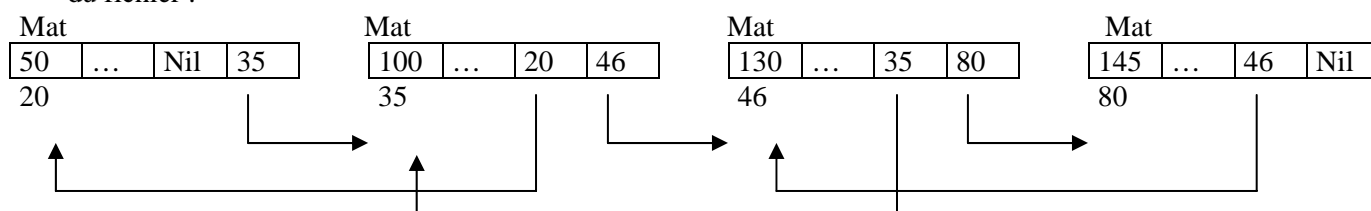


**Exercice 01 : Organisation séquentielle chaînée (A traiter par tous les groupes)**

Soit le fichier « Etudiant » dont le descripteur est le suivant :

*Etudiant{Matricule,Nom,Prénom,Adresse}*

- 1- En posant des hypothèses sur les rubriques du fichier, quelle sera la longueur d'un article ?
- 2- Si le fichier contient une vingtaine d'articles alors quelle sera la taille du fichier si les articles sont de longueur fixe ?
- 3- On voudrait organiser ce fichier en séquentiel chaîné. Chaque nœud peut contenir un article. Voici l'état initial du fichier :



La table de disponibilité contient les adresses libres suivantes : 30,60,70, 95 et 98.

On voudrait insérer les articles dont les valeurs de matricules sont : 60,110,120 et 138. Donner l'état du fichier après insertion de ces articles.

4 - On voudrait maintenant organiser ce fichier en un ensemble de blocs ou pages chaînées en avant. Discuter la structure du bloc selon que les enregistrements seront de longueur fixe ou variable. Quels sont les avantages et les inconvénients de chacune des approches ? Réorganiser l'état du fichier précédent selon cette approche.

5 – Considérons maintenant que chaque page peut contenir 5 enregistrements de longueur fixe « l ». Un bloc spécial appelé répertoire contient le nom du fichier et l'adresse de sa première page. La dernière page du fichier pointe vers le répertoire.

- Représenter cette organisation par un schéma pour plusieurs fichiers présents sur le support.

6 – Décrire l'algorithme de recherche en citant tous les cas possibles (y compris les débordements).

7 – Quelle est l'adresse d'un enregistrement ?

**Exercice 02 : Organisation séquentielle indexée (Groupes 1 & 3)**

On voudrait enregistrer les articles suivants dans un fichier séquentiel indexé utilisant un index primaire non dense sur le matricule.

Matricule	Nom	Autres Infos
14	Ali	...
25	Omar	...
10	Kamel	...
36	Ali	...
15	Wafaa	...
09	Nora	...
24	Lotfi	...

Matricule	Nom	Autres Infos
11	Lies	...
30	Amine	...
33	Nesrine	...
20	Malek	...
02	Hind	...
07	Lynda	...
18	Zoheir	...

On donne les paramètres suivants :

- Taille d'une page physique = 256 Octets
- Taille d'un article = 80 Octets
- Longueur Matricule = 10 Octets
- Longueur Nom = 22 Octets
- Une adresse = 2 Octets

### Questions :

- 1 – Donner le nombre d'enregistrements logiques par enregistrement physique.
- 2 - Combien de pages physiques seront-elles nécessaires pour tous ces articles ?
- 3 - Construire le fichier contenant ces articles selon l'organisation séquentielle indexée ainsi que l'index primaire construit sur la rubrique « Matricule ».
- 4 – On voudrait maintenant construire pour le même fichier un index secondaire sur la rubrique « Nom ». Construire cet index en expliquant.
- 5 – On voudrait insérer l'article « 17, Soulaf, ... ». Quel sera l'effet sur les index primaire et secondaire ?

### Exercice 03 : Organisation ISAM (Groupes 2 & 4)

Soit l'état Initial du fichier F suivant :

Index Piste	→	P <sub>0</sub>				
		P <sub>1</sub>	3	7	10	12
Zone primaire	→	P <sub>2</sub>	18	20	21	22
		P <sub>3</sub>	27	29	30	33
ZDC	→	P <sub>4</sub>	24 P <sub>1-1</sub>	36 P <sub>2-4</sub>		26 Nil

Cylindre 01

P <sub>0</sub>			
P <sub>1</sub>	40	43	48 50
P <sub>2</sub>	56	57	63 66
P <sub>3</sub>	69	74	
P <sub>4</sub>	67 Nil	53 P <sub>1-2</sub>	52 P <sub>4-2</sub>

Cylindre 02

Le cylindre 3 est utilisé comme zone de débordement indépendante dont la piste P0 correspond à l'index Cylindre du fichier :

Index Cylindre	→	P <sub>0</sub>			...
		P <sub>1</sub>	25 P <sub>4-4</sub>	54 Nil	
ZDI	→	P <sub>2</sub>		35 P <sub>4-2</sub>	37 Nil
		P <sub>3</sub>			
		P <sub>4</sub>			

Cylindre 02

### Questions :

1. Donner le nom de cette organisation.
2. Compléter les index pistes et l'index cylindre de ce fichier.
3. Quel sera l'état du fichier après insertion de l'article de clé 23 ?
4. Que se passera-t-il si l'on veut insérer l'article de clé 99 ? (Expliquer le procédé).
5. En sachant qu'une adresse nécessite 2 octets, qu'une clé en nécessite 2 aussi et qu'un article a une longueur de 110 octets, calculer :
  - L'espace occupé par les zones primaires du fichier.
  - L'espace occupé par les index de piste et de cylindre.

#### **Exercice 04 : Organisation séquentielle chaînée paginée avec index (A faire chez vous)**

Soit un système de gestion de fichiers qui permet l'organisation séquentielle chaînée à pages où on utilise un répertoire disque dont chaque entrée sert à retrouver la première page de chaque fichier séquentiel chaîné.

L'en-tête de chaque page d'un fichier de données contient : le chaînage avant, le chaînage arrière et le compteur d'articles dans la page.

Les pages libres du disque forment un fichier ZDD (Zone de Disponibilité du Disque) organisé lui aussi en séquentiel chaîné.

La taille d'une page (unité de transfert) est de 1024 octets, une adresse occupe 3 octets et un compteur est sur 3 octets.

Le fichier ABONNES a été stocké sur disque en séquentiel chaîné. Les articles sont de longueur fixe, la clé occupe 10 octets et on peut stocker au maximum 5 articles par page. Le descripteur du fichier ABONNES est le suivant :

***ABONNES{Code\_Compteur, Nom, Adresse, Catégorie, Index\_Compteur, Dette\_Antérieure}***

Soit (E0) l'état initial du disque (répertoire, ZDD, ABONNES) donné en figure (voir document en annexe).

#### **Questions :**

##### **Partie I :**

1. Déterminer la taille maximale que peut avoir un article ABONNES.
2. Donner l'état (E1) du disque après insertion de l'abonné de clé 091.
3. A partir de l'état (E1), donner l'état (E2) du disque après suppression de l'abonné de clé 040.
4. De manière générale expliquer ce qui se passera si l'insertion aura lieu entre 2 pages dont la première est pleine et la seconde pas.
5. Même question pour 2 pages complètement pleines.
6. L'insertion sera-t-elle possible si ZDD est vide ? (expliquer les différentes situations possibles)

##### **Partie II :**

Pour accélérer les accès aux articles du fichier ABONNES, on décide de rajouter une méthode d'accès supplémentaire utilisant un index non dense sur ce même fichier. On reprend l'exercice à partir de l'état (E0).

1. Quelle est la taille de l'index (en nombre de pages) ?
2. Donner l'état (F0) représentant l'état initial du disque dans cette nouvelle organisation.
3. Donner l'état (F1) du disque après l'insertion de l'abonné de clé 091.
4. A partir de l'état (F1), donner l'état (F2) du disque après suppression de l'abonné de clé 040.

**Remarque** : pour chaque état demandé, représenter seulement les parties ayant changé si vous le souhaitez.



**Bon Travail**

ORGANISATION SEQUENTIELLE CHAINEE A PAGES : ETAT INITIAL (E<sub>0</sub>) (POUR L'EXERCICE 4 DE LA SERIE TD3)

ZDD :

@ de la page suivante	250	100	300	400	700	800	
@ de la page	450	250	100	300	400	700	800

REPERTOIRE DISQUE :

NOM FICHIER	PARAMETRES FICHIER	@ de la 1 <sup>ère</sup> page
ZDD		450

ABONNES 200

FICHIER DES ABONNES A L'ETAT (E<sub>0</sub>):

150		01	050	200	05	350	150	05		050	04
040			085			098				109	
			088			100				111	
			090			103				114	
			092			105				118	
			094			107					
200			150			050				35	



# **Question n° 6 de l'exercice n° 02 de la série TD3**

# Table des matières

## Introduction

## I - Section

A. Algorithme de recherche.....	5
B. Algorithme d'insertion.....	7
1. 1° cas : si le fichier n'est pas trié alors les articles seront stockés selon leur ordre d'arrivée.....	7
2. 2° cas : si le fichier est trié sur la clé primaire (ici, c'est la rubrique matricule), alors il faudra respecter l'ordre du tri à chaque insertion.....	8
	9

# Introduction

Enoncé de la question :

6 – Décrire l'algorithme de recherche en citant tous les cas possibles (y compris les débordements).

# Section

Algorithme de recherche

7

Algorithme d'insertion

7

## A. Algorithme de recherche

1. Charger le répertoire en mémoire
2. Rechercher l'entrée correspondante au nom du fichier
3. Si non trouvé alors signaler « Fichier Inexistant »
4. Sinon récupérer l'adresse du fichier
5. Charger la page en mémoire vive (RAM) et chercher l'article dans la page
6. Si trouvé alors « Afficher l'article » et arrêter la recherche
7. Sinon récupérer l'adresse de la page suivante du fichier
8. Répéter les étapes de 5 à 8 jusqu'à ce que l'article soit trouvé ou bien que la fin du fichier soit atteinte
9. Si la fin du fichier est atteinte et l'article n'est pas trouvé alors signaler « Article Inexistant »

## B. Algorithme d'insertion

Toute insertion ne se fera qu'après avoir effectué une recherche qui délivrera un résultat négatif, i-e, « Article Inexistant ».

### 1. 1° cas : si le fichier n'est pas trié alors les articles seront stockés selon leur ordre d'arrivée.

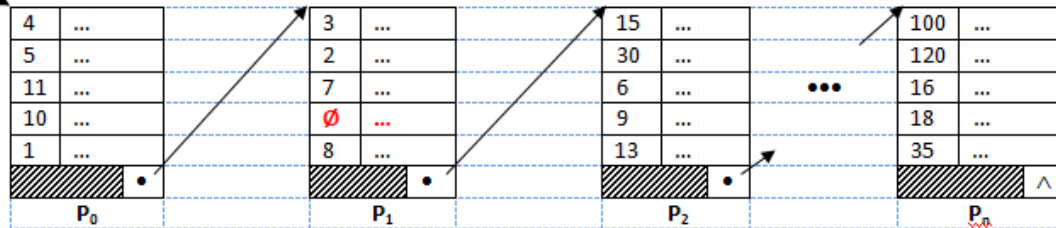
- ☐ Dans ce cas, l'insertion se fera au premier emplacement libre trouvé dans une page.
- ☐ Si toutes les pages du fichier sont pleines alors on allouera une nouvelle page qui sera chaînée à la dernière page en mettant à jour le pointeur de l'ancienne dernière page.



□ La nouvelle dernière page contiendra l'article inséré.

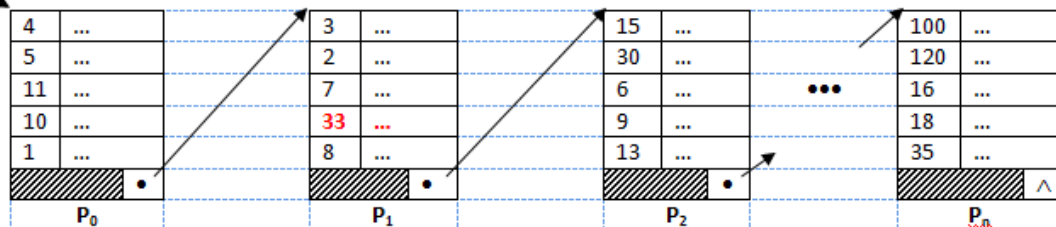
Voici un exemple où une page contient un emplacement vide :

• @début du fichier



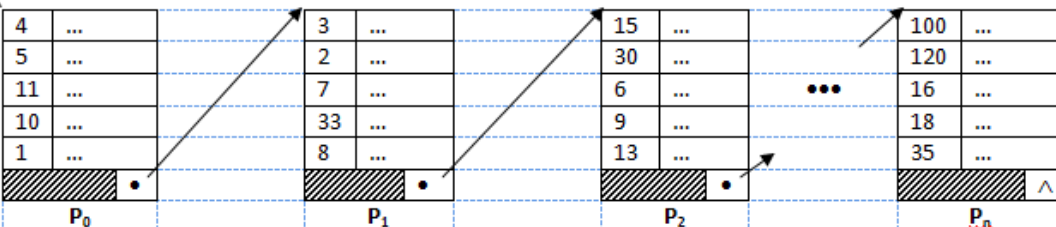
La page P1 contient un emplacement vide ( $\emptyset$ ). Nous allons y loger l'enregistrement de clé « 33 » par exemple. Ceci donnera le résultat suivant :

• @début du fichier



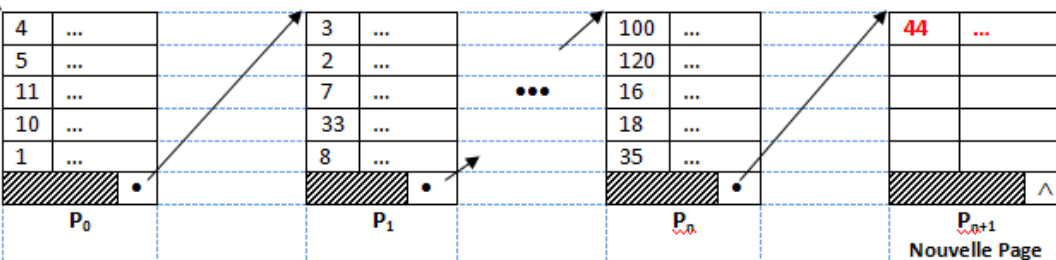
Voici un autre exemple où toutes les pages sont saturées, ce qui conduira à la création d'une nouvelle page pour y insérer l'article de clé "44" :

• @début du fichier



Etat après ajout d'une nouvelle page :

• @début du fichier



## 2. 2° cas : si le fichier est trié sur la clé primaire (ici, c'est la rubrique matricule), alors il faudra respecter l'ordre du tri à chaque insertion.

☐ Trouver la page où l'insertion doit avoir lieu

☐ Si la page où doit se faire l'insertion n'est pas pleine alors insérer l'article à l'endroit qu'il lui faut en décalant si nécessaire les autres articles dans la page.

Voici un exemple d'insertion de l'article de clé « 44 » :

• @début du fichier

1	...	6	...	11	...	30	...
2	...	7	...	13	...	33	...
3	...	8	...	15	...	35	...
4	...	9	...	16	...	100	...
5	...	10	...	18	...		
P <sub>0</sub>		P <sub>1</sub>		P <sub>2</sub>		P <sub>3</sub>	

Etat après insertion : décalage de l'article de clé « 100 » vers le bas.

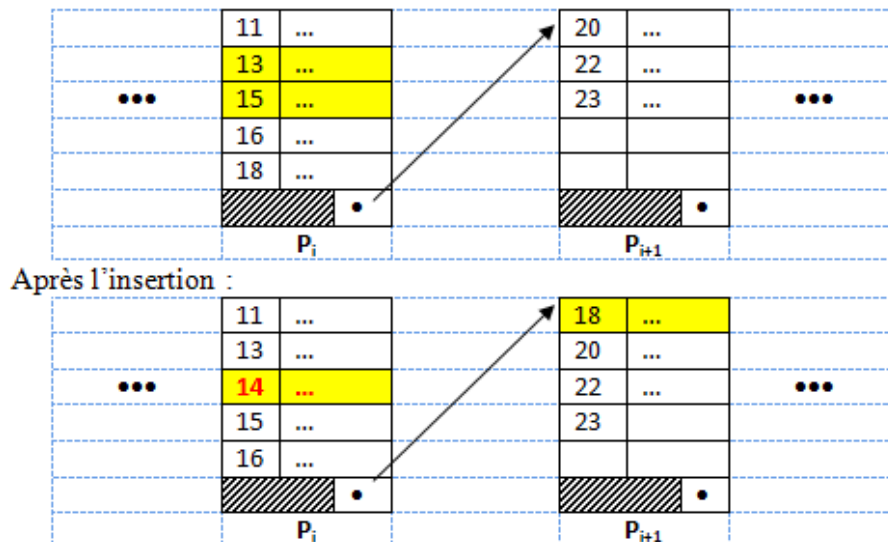
• @début du fichier

1	...	6	...	11	...	30	...
2	...	7	...	13	...	33	...
3	...	8	...	15	...	35	...
4	...	9	...	16	...	44	...
5	...	10	...	18	...	100	...
P <sub>0</sub>		P <sub>1</sub>		P <sub>2</sub>		P <sub>3</sub>	

☐ Si la page est pleine (saturée) alors il faudra voir l'état de la page suivante du fichier :

Si la page suivante n'est pas saturée alors insérer l'article à la position qu'il lui faut dans la page courante en débordant l'article éjecté vers la page suivante.

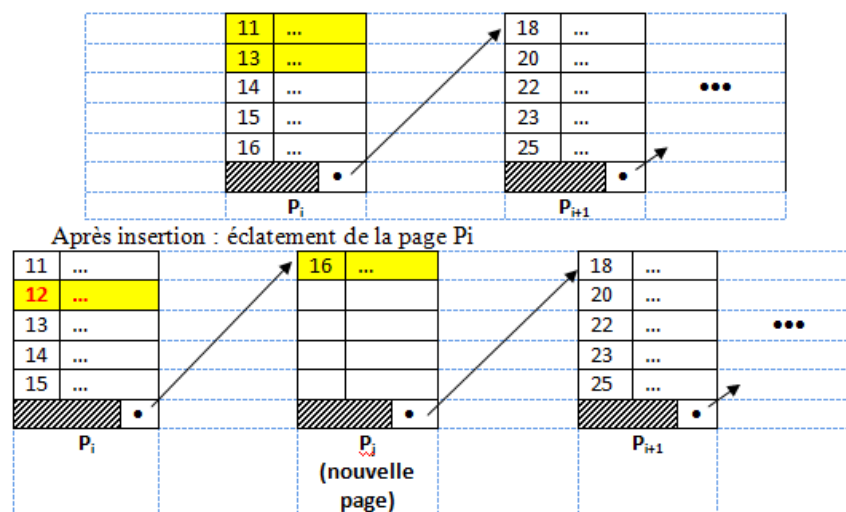
Voici un exemple : insertion de l'article de clé « 14 »



□ Si la page suivante est aussi saturée alors il faudra éclater la page courante, i-e, allouer une nouvelle page et y mettre l'article qui débordera après insertion.

□ Ensuite, il faudra mettre à jour l'adresse des pointeurs en chaînage avant.

Voici un exemple : insertion de l'article de clé « 12 »



□ Si la page où doit se faire l'insertion est la dernière page du fichier (i-e, il n'y a plus de page suivante) alors :

□ Si la clé de l'article à insérer est plus petite que la dernière clé de la page alors insérer l'article à la position qu'il faut en décalant. Si la page en question est saturée alors on allouera une nouvelle page qui recevra le dernier article qui sera expulsé vers elle. La nouvelle page deviendra la dernière du fichier et il faudra mettre à jour les chaînages.

Voici un exemple : Insertion de l'article de clé « 45 »

	30	...
	33	...
...	35	...
	44	...
	100	...
		^
	$P_n$	

Après insertion : l'article de clé « 100 » est expulsé vers la nouvelle page  $P_{n+1}$

	30	...	100	...
	33	...		
...	35	...		
	44	...		
	45	...		
				^
	$P_n$		$P_{n+1}$	

□ Si la clé à insérer est plus grande que la dernière clé de la page et que celle-ci est saturée alors allouer directement une nouvelle page et y insérer l'article puis mettre à jour les chainages.

Voici un exemple : Insertion de l'article de clé « 110 »

	30	...
	33	...
...	35	...
	44	...
	100	...
		^
	$P_n$	

Après insertion : l'article de clé « 110 » est directement placé dans la nouvelle page  $P_{n+1}$

	30	...	110	...
	33	...		
...	35	...		
	44	...		
	100	...		
				^
	$P_n$		$P_{n+1}$	

□ Notons enfin que le cas suivant peut aussi se présenter :

□ Si la page où doit se faire l'insertion est la première page du fichier alors il faudra vérifier si la clé de l'article à insérer est plus petite que toutes les clés de la page.

□ Si tel est le cas et que la page est saturée alors allouer une nouvelle page qui deviendra la première page du fichier et y placer l'article à insérer. Puis mettre à jour les chainages. De plus, mettre à jour l'adresse du fichier dans le répertoire bloc.

• @début du fichier

Nom Fic.	@ Fic.		4	...	
...	...		5	...	
			7	...	...
$F_i$	$@ F_i \bullet (P_1)$		10	...	
			11	...	
Répertoire Bloc			$P_1$		

Après insertion de l'article :  $P_0$  va devenir la nouvelle première page du fichier

• @début du fichier

Nom Fic.	@ Fic.		2	...	4	...
...	...				5	...
					7	...
$F_i$	$@ F_i \bullet (P_0)$				10	...
					11	...
Répertoire Bloc			$P_0$		$P_1$	

□ Si par contre la clé à insérer n'est pas la plus petite clé alors on regardera l'état de la page suivante et on procèdera comme expliqué dans les étapes précédentes.