

# BULLETIN D'INFORMATIQUE APPROFONDIE ET APPLICATIONS

COMPUTATION - INFORMATION

COMITE SCIENTIFIQUE

N° 59 - JUIN 2001

*Patrick Abellard  
Françoise Adreit  
Jalal Almhana  
France Chappaz  
M'hamed Charifi  
Roger Cusin  
Bernard Goossens  
Patrick Isoardi  
Robert Jacquier  
Jean - Philippe Lehmann  
Nadia Mesli  
Patrick Sanchez  
Rolland Stutzmann  
André Tricot*

DIRECTEUR

*Jean - Michel Knippel*

REDACTEUR EN CHEF

*Edmond Bianco*

REDACTEUR ADJOINT

*Sami Hilala*

SECRETARIAT

*Kalassoumi Adjilani*

Université de Provence  
Equipe Hermès. Case 33  
3, place Victor Hugo  
F - 13331 Marseille Cedex 3  
Téléphone: (0)4 91 10 62 30  
Télécopie : (0)4 91 50 91 10

DEPOSITAIRE

Université de Provence  
Bibliothèque Vniversitaire  
3, place Victor Hugo  
F - 13331 Marseille Cedex 3  
Téléphone: (0)4 91 62 44 16  
Télécopie : (0)4 91 95 75 57

**1 EDITORIAL.**

Informatique, économie et compagnie (suite 3),

*par Edmond Bianco*

**5 INTRODUCTION A L'ALGORITHMIQUE:  
LES FICHIERS,**

*par Jean - Louis Imbert*

**35 VOUZAVEDIBISAR.**

Animalcule Etrange,

*par Eric Olivier*

<http://scamup.univ-mrs.fr/biaa>

<http://www.up.univ-mrs.fr>

Publication trimestrielle, gratuite, de l'Université de Provence

Edition 2001

ISSN 0291 - 5413



**INFORMATIQUE, ECONOMIE et COMPAGNIE (suite 3).**

Nous avons l'outil, appliquons le. Il reste à voir si ça marche.

Le sport de compétition, dit de "haut niveau", est en fait un sport spectacle. Or, pour continuer à attirer les foules, c'est-à-dire faire du fric, il faut que les "athlètes" en donnent toujours plus. Il se trouve que la pharmacie est en mesure de gonfler un athlète, comme on gonfle un moteur de voiture de course. Mais comme pour le moteur, le gonflage n'est pas sans conséquences sur la vie du "gonflé". Deux axiomes s'appliquent à la suite, le 3) la communauté barbare met en coupe réglée le milieu du sport, et ensuite le 4) l'inertie fait que, même si le scandale du dopage éclate, on continue de trafiquer le corps humain, mais alors sous contrôle médical. Le dopage n'est plus un gonflage, mais un soin.

La relaxation s'applique aussi au sport spectacle. Dans les années cinquante soixante, c'était la boxe qui attirait les foules. Les trafics et abus divers ont tari la poule aux œufs d'or, le Noble Art ne fait plus salle comble. Maintenant c'est le foot, à une moindre mesure le tennis et le cyclisme qui permettent de faire des sous. Mais, il est assez facile de prévoir que le principe 2, de relaxation, va s'appliquer, et la vitrine commerciale sera obligée d'exposer de nouvelles attractions. Le cataclysme paraît inéluctable, sous forme d'un contrôle judiciaire, qui fera sortir ces spectacles du domaine de la rentabilité. Rentabilité qui n'est rien d'autre que le moteur, l'essence, de la communauté barbare. Un autre exemple surprenant de validité de l'axiome 4, l'inertie, est illustré par l'affaire de l'O.M., géré par Tapie. Tapie, mène l'O.M. à la victoire, supporté par une foule enthousiaste. Malheureusement, il est rapidement convaincu d'imposture, les victoires obtenues sont de fausses victoires bricolées. On aurait pu penser que le supporter marseillais, ainsi trompé, allait réagir "sportivement" et conspuer le ripoux. Eh bien pas du tout, le public est prêt à admettre tous les comportements pourvu qu'ils mènent à une victoire, même factice.

L'axiome 3, dit communauté barbare, pourrait porter d'autres noms, celui là ne fait que marquer l'origine du phénomène. A l'époque des grands découvreurs de richesses exotiques, quand ces richesses voyageaient par mer, on aurait pu parler d'écumeurs des mers, de corsaires, de pirates. Au temps de la Révolution, on disait fermiers généraux. De nos jours on pourrait dire rentabilité, plus value, profit; ou alors carrément requins. Comme on le voit, la palette est très large, et loin d'être épuisée.

L'axiome 7, la colique, peut être déterminant pour l'amorce d'une phase cataclysmique. Envisageons un petit scénario catastrophe, parfaitement imaginaire, donc. Le football est un spectacle qui rapporte des milliards, les clubs, les grands clubs, bien sur, ceux qui sont épaulés par de grosses entreprises, se montent en sociétés, bientôt si puissantes qu'elles se cotent en bourse. Mais, pour rapporter, il faut les meilleurs joueurs du monde. Alors on va fouiller dans les communautés sans espoir, les bidonvilles du quart monde. On fait briller à des jeunes sans avenir, une carrière à la Zidane. On les soumet à un concours drastique, et ces candidats n'ont de choix que réussir ou se retrouver, sans papiers, entre les mains de la police, puis dans un charter. C'est le cas de la plupart d'entre eux, qui seront expulsés manu militari. Les champions, les heureux vainqueurs, eux, vont être intéressés à la vie de leur club en devenant actionnaires. Mais en fonction des résultats, ces héros vont être rachetés par des clubs plus riches ou plus dynamiques, de telle sorte qu'ils risquent de devenir actionnaires de clubs adverses. Et le club qui perdra un peu trop va miner la confiance de ses financiers. Voilà une situation rêvée pour un magnifique krach. Car la chute d'un club risque d'en entraîner d'autres. Et l'effet colique joue alors à plein. Rassurons nous pourtant, les petits malins auront retiré leur épingle du jeu, bien avant.

L'axiome 4 de l'inertie est si puissant, épaulé par l'effet communauté barbare, que même les tentatives de critique du système les plus acerbes, les plus justifiées sont intégralement limées, ajustées et finalement récupérées. C'est le cas de nombre de slogans de Mai 1968, ou de positions des écologistes, qui servent maintenant à vendre des chaussettes, des slips, du dentifrice, ou des voitures. Cette curieuse capacité à reconverter même les pires critiques, montre deux propriétés: le système est stable, la critique, si virulente qu'elle puisse être, ne sort pas du système.

Cette surprenante stabilité est assurée par le jeu de trois axiomes. L'axiome informatique-abstraction isole l'individu, perdu au milieu des arcanes de plus en plus complexes de la civilisation. Le délit d'initié n'est pas une invention nouvelle, déjà du temps de la Rome antique ne disait-on pas que deux augures ne pouvaient se regarder sans rire! Nos politiciens modernes ont perdu le sens de l'humour à un point qu'ils peuvent parfaitement se regarder tout en continuant à se prendre au sérieux. Les singeries réussissent encore à en imposer au bon peuple.

Encore faut-il se renouveler un peu.

La passion du jeu, sans doute exacerbée par le sentiment d'exclusion, conséquence de l'axiome informatique-abstraction, laisse-t-elle au dedans de chacun de nous l'espoir de gagner un jour le gros lot: la femme idéale, le prince charmant, le poste de petit chef tant attendu, la gloire dans un métier de rêve ... La chanson d'Aznavor dit très bien ces choses: "Je m'voyais tout en haut de l'affiche ...".

La communauté barbare sait exploiter ces vains espoirs:

-les guerres les plus violentes opposent les gangs pour la maîtrise des commerces de l'illusion;

-le commerce de la drogue pour l'accession aux paradis artificiels;

-le commerce de la prostitution dont la clientèle est stabilisé par un autre commerce, celui des tabous sexuels, fonds de commerce de l'église;

-le commerce des jeux d'argent en tous genres, les casinos; ou même la simple et innocente Loterie Nationale.

L'inertie fait le reste, en effet pourquoi changer un monde si plein de promesses. Surtout qu'il y en a pour tous les goûts, on peut même y jouer les Robin des Bois avec plus ou moins de bonheur.

## *L'IMPOSTURE*

La communauté barbare a réussi un autre exploit extraordinaire. Non seulement elle s'est infiltrée dans la cité au point d'en écumer l'essentiel tout en ayant l'air d'en faire intimement partie, mais en plus elle a réussi à convaincre la communauté ordinaire que sans elle, rien ne pourrait avancer. Elle se pose en moteur du progrès.

Le pivot de l'opération se décrit en un mot: "trust", la confiance. Il faut faire confiance au chef. Il suffit d'observer les résultats d'une telle confiance, dans les opérations acrobatiques du Crédit Lyonnais, ou bien dans celles du célèbre ami de la France, le milliardaire Maxwell, dont on a appris l'usage qu'il avait fait de la caisse de retraite de ses trente mille employés, juste après son "suicide" marin. Les "responsables" du Crédit Lyonnais ne se sont pas suicidés, eux.

Les holdings sont dirigés par des gens qui ne connaissent strictement rien à ce qui se fait dans leurs innombrables entreprises, mais qui savent parfaitement comment utiliser, pour leur pouvoir personnel, le fruit du travail de leurs employés. On connaît les expressions:

" La science est une affaire bien trop sérieuse pour être confiée à des scientifiques ..." ou encore:

" Moi, Monsieur, vous voyez je suis obligé de faire de l'Administration, alors que j'aimerais bien mieux être dans mon laboratoire ..." Il aimerait certes, mieux être dans son "labo", mais c'est tellement plus amusant de fricoter avec les gens du pouvoir, on a tellement plus l'air d'être un personnage important.

Avoir de l'entregent.

Telle Janus, l'administration a deux visages. L'administration traditionnelle, réactionnaire, qui s'oppose à tout mouvement en introduisant une viscosité maximale dans les mouvements de la société. Et la partie d'administration

investie par la communauté barbare qui introduit une étrange dynamique, brûlant des forêts pour pouvoir investir des terrains interdits à la construction, délocalisant les entreprises vers des pays sans lois, répandant une dangereuse chimie à tort et à travers, etc...

## *UTOPIE*

Toute conception d'un monde différent, plus adapté à la vie et aux souhaits des gens fait partie du domaine de l'utopie. Par définition. Changer le monde revient de toute évidence à changer les axiomes qui servent de brique pour bâtir la société. La difficulté provient de la nature même de ces axiomes, qui sont les conséquences de la psychologie des masses. Or, que sont les masses sinon constituées d'êtres ayant subi, souvent à l'insu de leurs instructeurs, les parents en général, une déformation quasi définitive qui les rend inaptés à remettre en question l'essentiel de l'ordre établi. Les composants des masses sont devenus des êtres frileux qui se sentent mal à l'aise à la seule évocation d'une situation vraiment étrangère à celle dans laquelle ils baignent habituellement. Et si, par hasard, ils se laissent entraîner dans une aventure qui déborde largement de la psychologie habituelle, c'est toujours à l'abri d'un chef qui sert de paratonnerre psychologique. C'était bien le cas des vieux grognards qui se faisaient hacher menu derrière Napoléon, des révolutionnaires qui suivaient le Che, des esclaves qui constituaient l'armée de Spartacus, et de tant d'autres. Emportés dans le même tourbillon, les chefs, quant à eux, tous aussi désorientés que leurs subordonnés, se réfugient dans un univers imaginaire qui leur permet de tenir le coup, mais gare s'ils réussissent leur entreprise, l'éducation reprend alors le dessus et on assiste à la naissance d'un nouveau tyran. Dont Fidel, le Danube de la Pensée Révolutionnaire, Staline et tant d'autres, qui réappliquent ce qu'ils ont appris dans leur enfance: la méfiance, la peur, la défiance, la punition, l'exclusion ...

## *BREF*

Bref; on pourrait essayer d'imaginer les axiomes d'une société, sinon parfaite, du moins, moins gravement déficiente, nombreux sont ceux qui s'y sont essayés, mais comment les appliquer si le support psychologique des masses, les rejette spontanément? Pourtant, à force de baigner dans la grisaille quotidienne, un petit voyage au Royaume d'Utopie reste bien tentateur.

*Edmond Bianco*

# INTRODUCTION A L'ALGORITHMIQUE: LES FICHIERS

*Jean - Louis Imbert*

imbert@iut.u-clermont1.fr

*Après les notions présentées dans le numéro précédent de mars 2001: structure d'un ordinateur, notions de base et tableaux, nous passons aux fichiers. Nous entrons de plein pied dans les modèles de données d'un ordinateur.*

*Nous sommes dans les longs flots d'objets d'un type donné. Il faut les organiser et pas n'importe comment en fonction de ce que l'on fait. Jean - Louis Imbert présente ici les fichiers séquentiels. Ensuite, le lecteur sera prêt pour attaquer les fichiers et systèmes de fichiers qui permettent d'organiser la mémoire secondaire pour améliorer l'efficacité des accès.*

*Présentation de Jean - Michel Knippel*

## **AVANT - PROPOS (rappel du numéro 58 de mars 2001)**

Notre but est de faire une introduction succincte à l'algorithmique. C'est à dire, de voir les notions de base du fonctionnement des algorithmes. Dans cette optique, nous enlèverons toute contrainte due à un langage de programmation particulier, et exprimerons les algorithmes en langages naturel. Ainsi, pour les applications futures, nous aurons le libre choix du langage de programmation, et pourrons prendre le plus adéquat pour l'algorithme concerné, et en fonction du travail à réaliser.

Pour mieux comprendre l'esprit des algorithmes, il nous a semblé préférable de donner auparavant, quelques notions du fonctionnement d'un ordinateur (chapitre 1), après quoi, nous introduisons les notions de base de l'algorithmique (chapitre 2 et compléments dans le chapitre 3). Les chapitres 3 et 4 sont respectivement consacrés aux notions et algorithmes de base, sur le fonctionnement des tableaux et des fichiers séquentiels.

## 4. LES FICHIERS

- 4.1. Définitions
- 4.2. Les fichiers séquentiels
- 4.3. Opérations sur les fichiers
- 4.4. Algorithme de lecture
- 4.5. Exercices
- 4.6. Algorithme d'ajout
- 4.7. Algorithme de correction
- 4.8. Exercices
- 4.9. Algorithme de suppression
- 4.10. Algorithme d'ajout dans un fichier trié
- 4.11. Exercices

### 4.1. DEFINITIONS

**DEFINITION:** *Un fichier est un ensemble d'informations résidant en mémoires secondaires (bande, disque, disquette...), pouvant donc atteindre de grandes tailles. Chaque fichier porte un nom.*

Il y a plusieurs types de fichiers, ceux qui nous intéressent ici, sont les fichiers de données.

**DEFINITION:** *Les informations d'un fichier de données sont regroupées en petits sous ensembles appelés enregistrements. Un enregistrement a une structure bien définie. Tous les enregistrements d'un même fichier de données ont la même structure.*

---

**Exemple:** Le fichier de nom "ETUDIANT" contient les noms des étudiants, et divers renseignements les concernant. Il est constitué d'enregistrements à raison d'un par étudiant.



Chaque enregistrement à la structure suivante:

- le nom de l'étudiant,
- son prénom,
- sa date de naissance,
- son adresse,
- son numéro de sécurité sociale,
- son numéro de carte d'étudiant,
- son année d'étude,
- le diplôme obtenu qui permet de l'inscrire.

ENREGIS- -TREMMENT	MARTIN	DUPONT	DURAND	
	Paul 22/08/95 rue Verlaine 1650813055124 12345A 3 deug	Jacques 01/01/68 place Vian 1680113055987 98765Z 1 bac	Alice 15/05/67 rue Voltaire 2670513168245 19283H 3 deug	-----

#### 4.2. LES FICHIERS SEQUENTIELS

Il y a plusieurs sortes de fichiers de données :

- \* Les fichiers **séquentiels** (on accède aux enregistrements l'un après l'autre, dans un ordre déterminé au moment de la saisie).
- \* Les fichiers à **accès direct** (on va directement à l'enregistrement désiré, à condition de connaître sa localisation).
- \* Les fichiers **séquentiels indexés** qui bénéficient des deux avantages précédents.

Nous ne nous intéressons qu'au cas des fichiers de type séquentiel. Dans ces fichiers on peut accéder à un enregistrement, uniquement en partant du premier enregistrement du fichier, et en parcourant tout le fichier jusqu'à l'enregistrement désiré. (Actuellement, certains systèmes permettent de parcourir le fichier également à reculons).

Outres les enregistrements, un fichier séquentiel possède une **marque de début de fichier**, et une **marque de fin de fichier**, qui permettent de le délimiter.

### 4.3. OPERATIONS SUR LES FICHIERS

Nous considérons que l'ordinateur sait faire les actions suivantes sur les fichiers séquentiels:

**Ouvrir un fichier en lecture,**

**Ouvrir un fichier en écriture,**

**lire un enregistrement dans un fichier,**

**écrire un enregistrement dans un fichier,**

**réécrire un enregistrement dans un fichier,**

**détecter la fin d'un fichier,**

et **fermer un fichier.**

*DÉFINITION: On appelle **Tampon** une zone en mémoire centrale (R.A.M), ayant la structure d'un enregistrement. Une zone tampon est toujours associée à un fichier précis.*

Il y aura donc autant de zones tampon que de fichiers ouverts.

Ouvrir un fichier crée une liaison Tampon-fichier.

Ouvrir un fichier en lecture, consiste, pour l'ordinateur, à chercher où se trouve le fichier, et à positionner le pointeur d'enregistrement sur le premier enregistrement du fichier. Si le fichier est vide, il pointera la marque de fin de fichier.

Ouvrir un fichier en écriture, consiste, pour l'ordinateur, à chercher où se trouve le fichier, et à positionner le pointeur d'enregistrement sur la marque de fin de fichier.

Pour lire dans un fichier, ce dernier doit être ouvert en lecture. L'action de lire consiste alors pour l'ordinateur à:

- 1) Récupérer dans la mémoire externe, le contenu de l'enregistrement pointé, pour le fichier considéré,
- 2) à le mettre dans la zone tampon associée à ce fichier, et qui se trouve en mémoire centrale,
- 3) enfin, à positionner le pointeur d'enregistrement de ce fichier, sur l'enregistrement suivant, ou la marque de fin de fichier s'il n'y en a pas.

Noter que si le pointeur d'enregistrement est sur la marque de fin de fichier, lorsqu'on demande de lire, l'ordinateur donnera un message d'erreur, car il ne pourra pas réaliser cette lecture.

Réécrire un enregistrement consiste à remplacer le dernier enregistrement lu, par le contenu du tampon. Le fichier doit donc à cet instant être ouvert en

lecture, et n'avoir pas été fermé depuis la dernière lecture. A noter qu'on ne peut pas réécrire si rien n'a été encore lu.

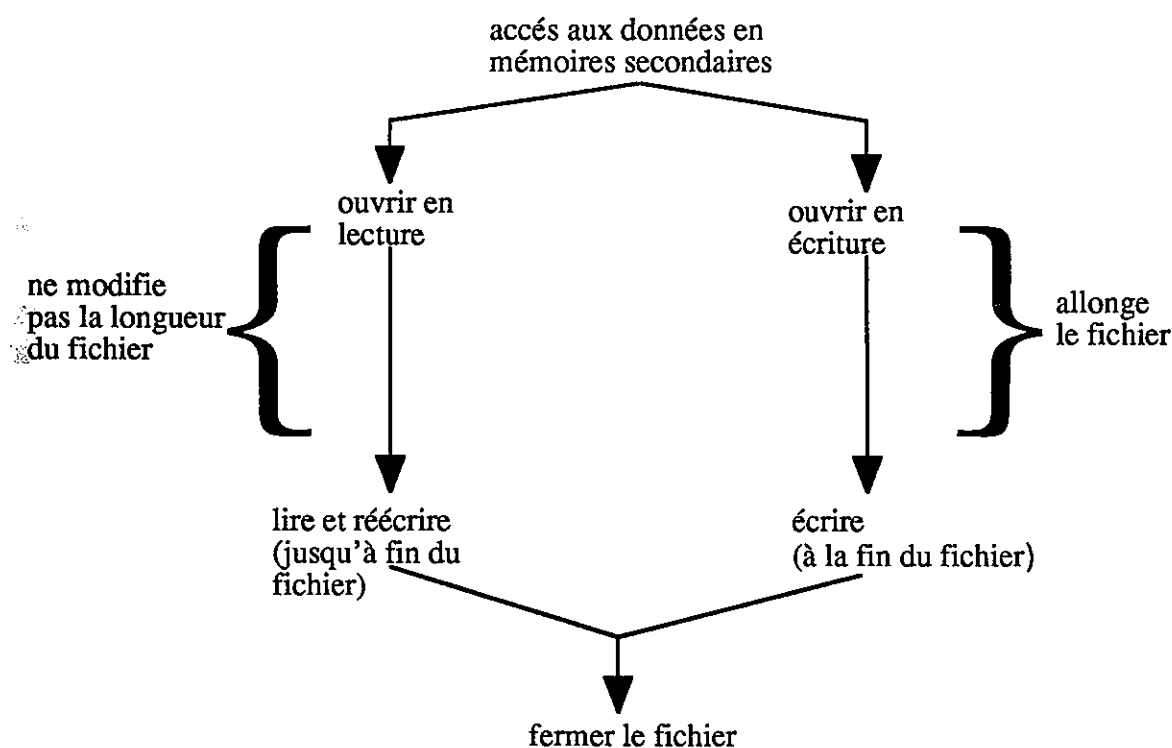
Pour écrire un enregistrement dans un fichier, ce dernier doit avoir été ouvert en écriture. L'action d'écrire consiste alors pour l'ordinateur à:

1) ajoute un nouvel enregistrement à la fin du fichier, la marque de fin de fichier étant repoussée de la longueur de cet enregistrement,

2) et à remplir le nouvel enregistrement avec le contenu du tampon associé à ce fichier.

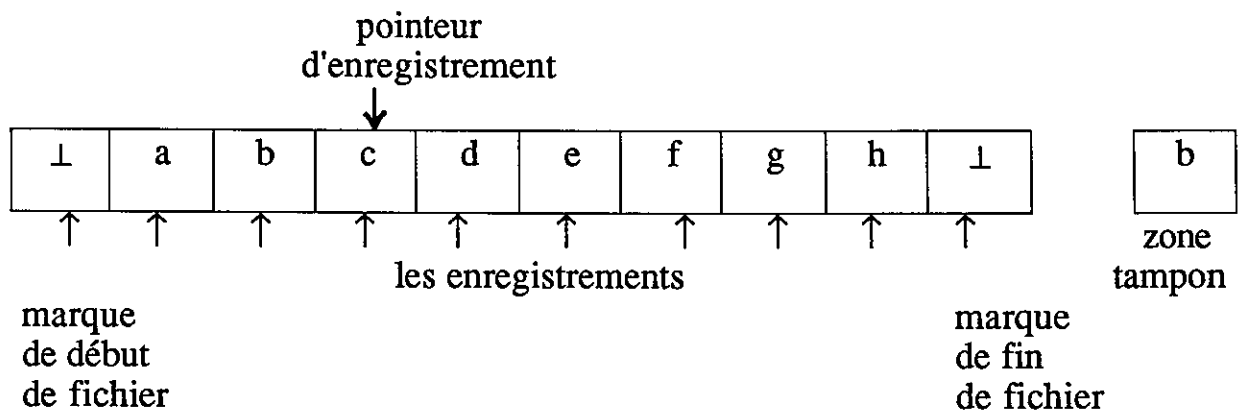
L'ordinateur détecte la fin du fichier dès que le pointeur d'enregistrement est positionné sur la marque de fin de fichier.

Fermer un fichier a pour effet de libérer la liaison tampon-fichier ainsi que le pointeur d'enregistrement, et de mettre à jour, dans la zone réservée à cet effet pour ce fichier, tous les renseignements le concernant. Il faut toujours fermer un fichier avant de terminer une application, sous peine de perdre le travail effectué, voire le fichier tout entier.



Pour des besoins pratiques et de démonstration, nous noterons pour un fichier f: f- l'ensemble des enregistrements qui précèdent strictement le pointeur

d'enregistrement, et  $f^+$  l'ensemble de ceux qui le suivent, y compris celui qui est pointé. Par exemple:



Dans ce cas,  $f^- = \{a,b\}$  et  $f^+ = \{c,d,e,f,g,h\}$ .

#### 4.4. ALGORITHME DE LECTURE

“Lire un fichier dont les enregistrements contiennent le nom et l’âge d’un ensemble d’individus, et écrire le nom et l’âge de ces individus.”

Précondition: Le fichier s’appelle “PERSONNEL”,  
un enregistrement est composé d’un nom et d’un âge.

Postcondition: Ecrire tous les noms et les âges correspondants.

Algorithme:

Début:

Ouvrir le fichier “PERSONNEL” en lecture.

Tant que la fin du fichier “PERSONNEL” n’est pas détectée, répéter l’opération suivante:

    début

        lire un enregistrement dans le fichier “PERSONNEL”,  
        écrire à l’écran le nom et l’âge

    fin.

Fermer le fichier “PERSONNEL”.

Fin.

ou en style abrégé:

Début :

Ouvrir-en-lecture ("PERSONNEL").

Tant que non fin-du-fichier ("PERSONNEL"), répéter :  
début

lire ("PERSONNEL"),  
écrire-écran (nom, âge)

fin.

Fermer ("PERSONNEL").

Fin.

Le fichier "PERSONNEL" n'a pas varié au cours de l'opération.

preuve:

Il y a un nombre fini d'enregistrements dans un fichier, il y a une seule boucle par enregistrement, l'algorithme se termine donc après un nombre fini de boucles.

Soit l'invariant "PERSONNEL<sup>-</sup> est écrit, et PERSONNEL<sup>+</sup> n'est pas écrit".

Il est vrai, juste après l'ouverture, car PERSONNEL<sup>-</sup> = {} et PERSONNEL<sup>+</sup> = PERSONNEL.

Au cours d'une application de l'opération décrite dans la boucle, le pointeur avance d'un enregistrement, après avoir mis dans la zone tampon le contenu de l'enregistrement qu'il pointait précédemment. Or, il écrit dans cette même boucle le contenu de ce tampon nouvellement chargé, donc si tous les enregistrements qui précédaient l'ancienne position du pointeur étaient écrits à l'écran avant l'entrée dans la boucle, tous les enregistrements précédents la position actuelle du pointeur sont bien écrit à l'écran. D'où PERSONNEL<sup>-</sup> est bien entièrement écrit à l'écran. Le reste de PERSONNEL n'étant pas intervenu, n'est toujours pas écrit. A la fin de l'algorithme, avant la fermeture, le pointeur étant sur la marque de fin fichier, PERSONNEL<sup>-</sup> = PERSONNEL, et PERSONNEL<sup>+</sup> = {}.

L'algorithme est donc acceptable.

---

Pour fixer les idées, nous allons faire une trace de l'exécution sur le fichier suivant:

⊥	Paul 22	Georges 21	Nathalie 18	Sylvain 18	Robert 19	Valérie 21	⊥
---	------------	---------------	----------------	---------------	--------------	---------------	---

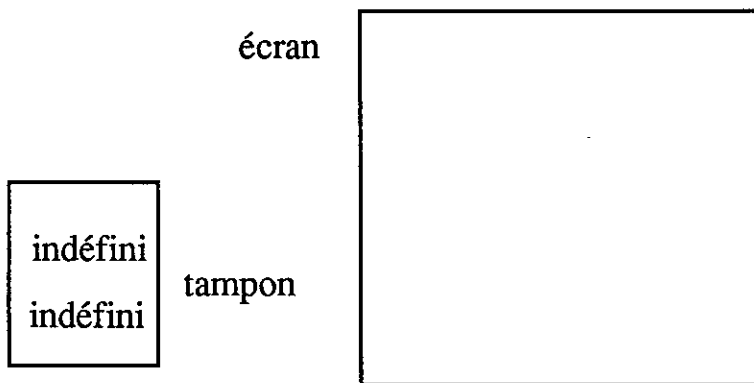
Au départ de l'algorithme, l'écran est vide, et le tampon est indéfini.

ouvrir-en-lecture ("PERSONNEL") provoque:

la réservation et l'association d'un tampon au fichier "PERSONNEL", ayant la même structure que les enregistrements du fichier, l'écran reste vide. Le tampon est soit vide, soit contient des caractères non contrôlés.

⊥	Paul 22	Georges 21	Nathalie 18	Sylvain 18	Robert 19	Valérie 21	⊥
---	------------	---------------	----------------	---------------	--------------	---------------	---

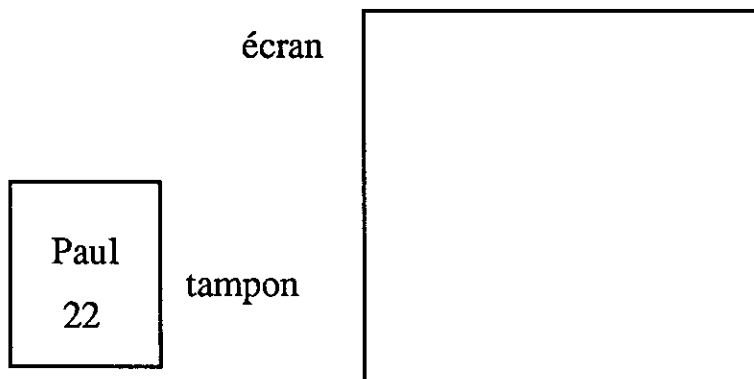
↑  
pointeur  
d'enregistrement



boucle 1: lire ("PERSONNEL")

⊥	Paul 22	Georges 21	Nathalie 18	Sylvain 18	Robert 19	Valérie 21	⊥
---	------------	---------------	----------------	---------------	--------------	---------------	---

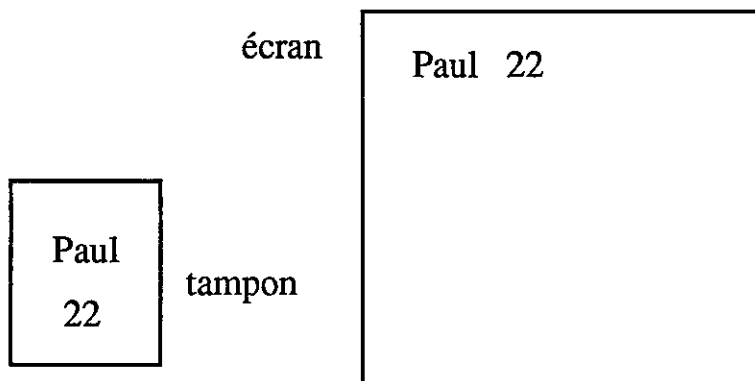
↑  
pointeur  
d'enregistrement



et écrire-écran (nom, âge)

⊥	Paul 22	Georges 21	Nathalie 18	Sylvain 18	Robert 19	Valérie 21	⊥
---	------------	---------------	----------------	---------------	--------------	---------------	---

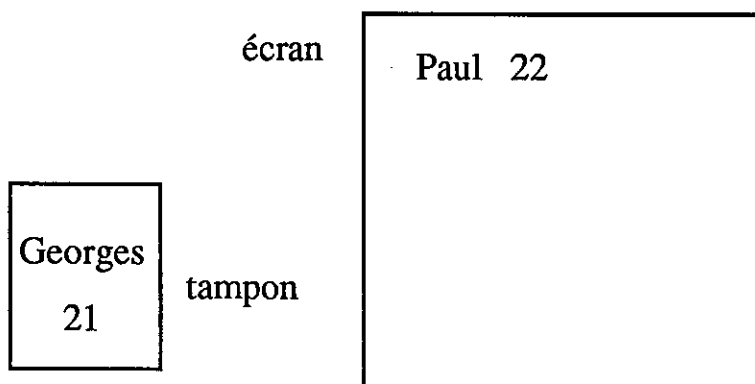
↑  
pointeur  
d'enregistrement



boucle 2: lire ("PERSONNEL")

⊥	Paul 22	Georges 21	Nathalie 18	Sylvain 18	Robert 19	Valérie 21	⊥
---	------------	---------------	----------------	---------------	--------------	---------------	---

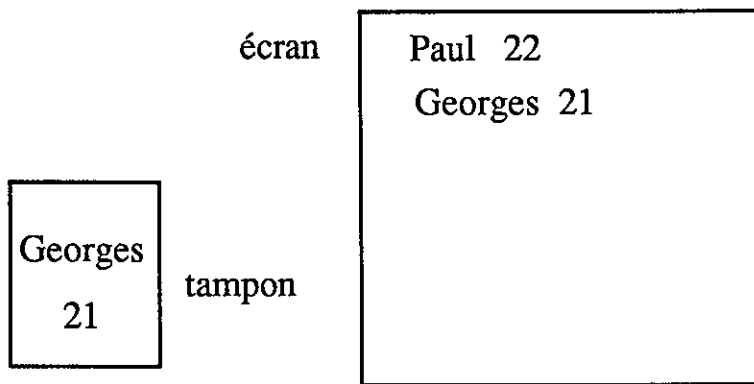
↑  
pointeur  
d'enregistrement



et écrire-écran (nom, âge)

⊥	Paul 22	Georges 21	Nathalie 18	Sylvain 18	Robert 19	Valérie 21	⊥
---	------------	---------------	----------------	---------------	--------------	---------------	---

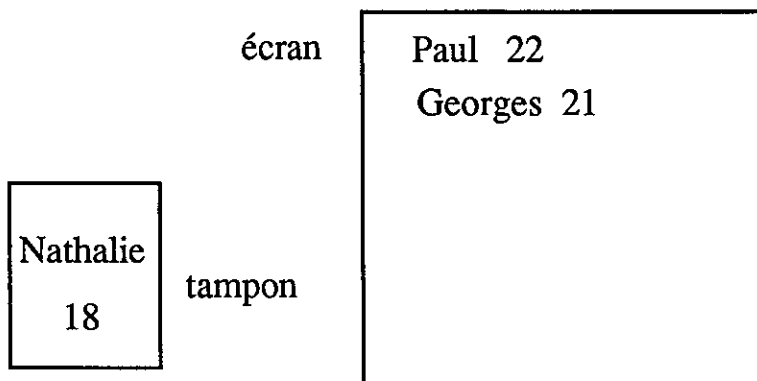
↑  
pointeur  
d'enregistrement



boucle 3: lire ("PERSONNEL")

⊥	Paul 22	Georges 21	Nathalie 18	Sylvain 18	Robert 19	Valérie 21	⊥
---	------------	---------------	----------------	---------------	--------------	---------------	---

↑  
pointeur  
d'enregistrement

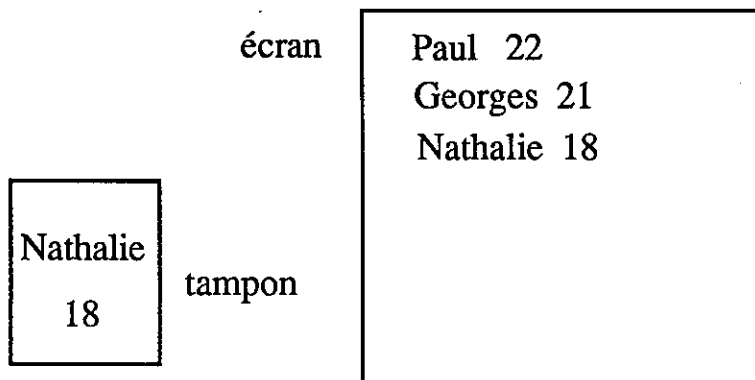




et écrire-écran (nom, âge)

⊥	Paul 22	Georges 21	Nathalie 18	Sylvain 18	Robert 19	Valérie 21	⊥
---	------------	---------------	----------------	---------------	--------------	---------------	---

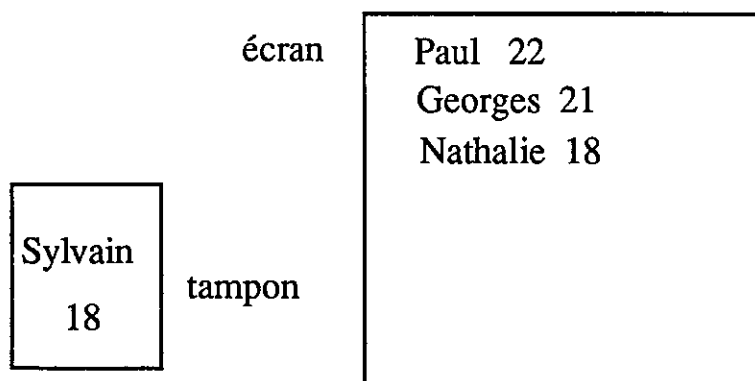
↑  
pointeur  
d'enregistrement



boucle 4: lire ("PERSONNEL")

⊥	Paul 22	Georges 21	Nathalie 18	Sylvain 18	Robert 19	Valérie 21	⊥
---	------------	---------------	----------------	---------------	--------------	---------------	---

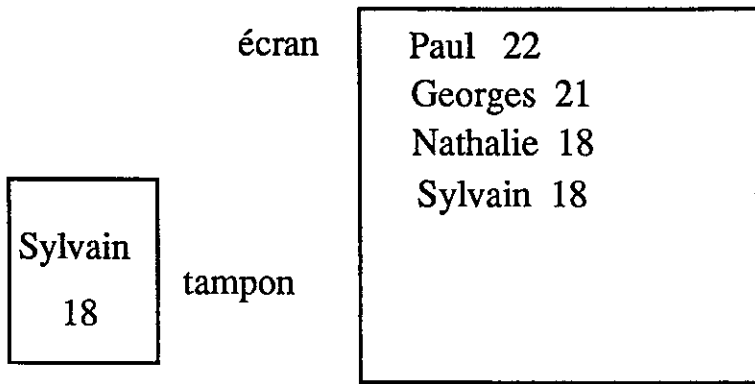
↑  
pointeur  
d'enregistrement



et écrire-écran (nom, âge)

⊥	Paul 22	Georges 21	Nathalie 18	Sylvain 18	Robert 19	Valérie 21	⊥
---	------------	---------------	----------------	---------------	--------------	---------------	---

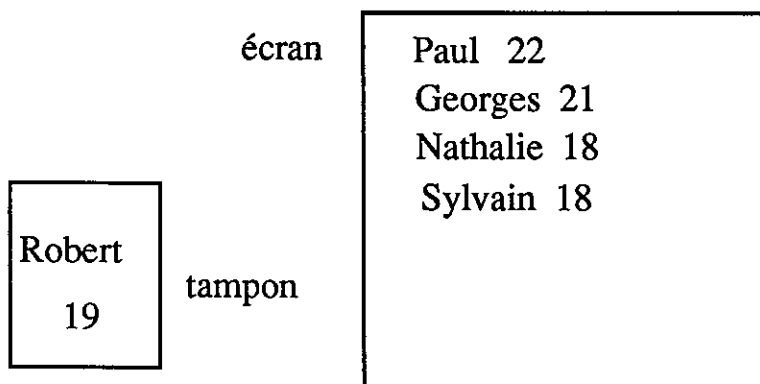
↑  
pointeur  
d'enregistrement



boucle 5: lire ("PERSONNEL")

⊥	Paul 22	Georges 21	Nathalie 18	Sylvain 18	Robert 19	Valérie 21	⊥
---	------------	---------------	----------------	---------------	--------------	---------------	---

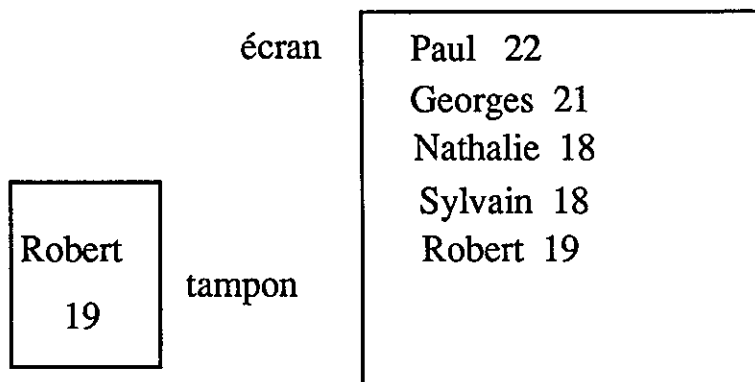
↑  
pointeur  
d'enregistrement



et écrire-écran (nom, âge)

⊥	Paul 22	Georges 21	Nathalie 18	Sylvain 18	Robert 19	Valérie 21	⊥
---	------------	---------------	----------------	---------------	--------------	---------------	---

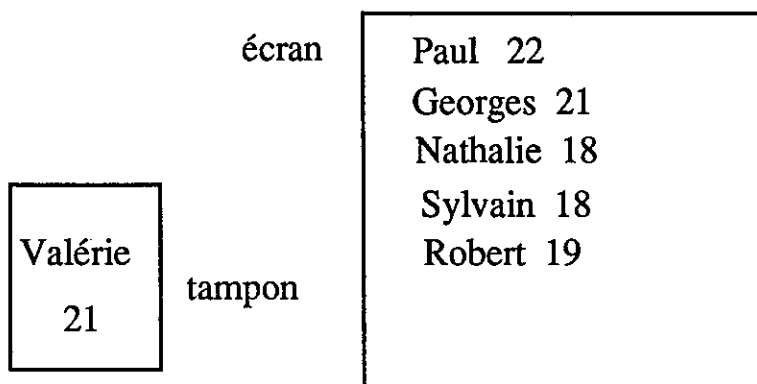
↑  
pointeur  
d'enregistrement



boucle 6: lire ("PERSONNEL")

⊥	Paul 22	Georges 21	Nathalie 18	Sylvain 18	Robert 19	Valérie 21	⊥
---	------------	---------------	----------------	---------------	--------------	---------------	---

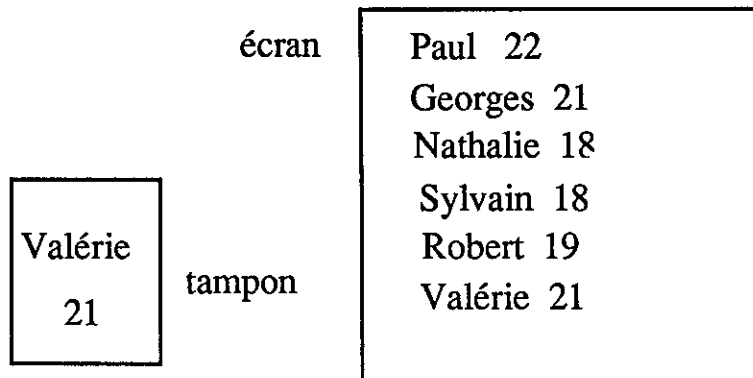
↑  
pointeur  
d'enregistrement



et écrire-écran (nom, âge)

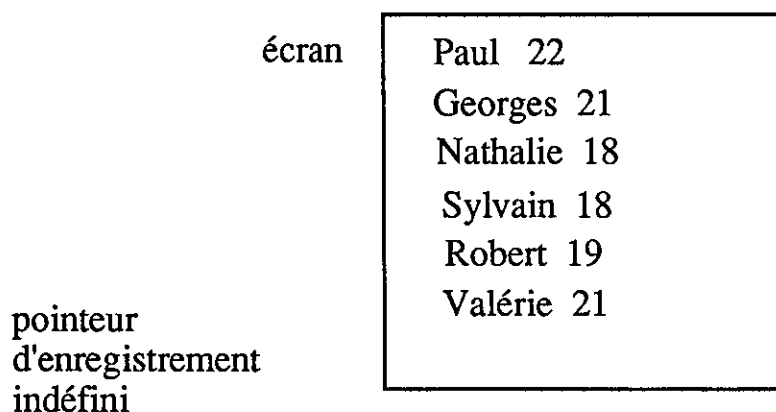
⊥	Paul 22	Georges 21	Nathalie 18	Sylvain 18	Robert 19	Valérie 21	⊥
---	------------	---------------	----------------	---------------	--------------	---------------	---

↑  
pointeur  
d'enregistrement



Ici, la fin du fichier ayant été rencontrée, la boucle est abandonnée. On passe alors à l'instruction suivante: fermer ("PERSONNEL")

⊥	Paul 22	Georges 21	Nathalie 18	Sylvain 18	Robert 19	Valérie 21	⊥
---	------------	---------------	----------------	---------------	--------------	---------------	---



Le tampon est dissocié du fichier, et le pointeur d'enregistrement ne pointe plus vers le fichier.

Fin de l'exécution.

#### 4.5. EXERCICES

- 1) Ecrire un algorithme qui compte le nombre d'enregistrements d'un fichier.

Dans la suite, les enregistrements du fichier "ETUDIANT" contiennent le nom, le sexe, l'âge et l'année d'étude des étudiants de la faculté.

- 2) Ecrire un algorithme donnant l'âge moyen des étudiants.
- 3) Ecrire un algorithme qui donne le nombre d'hommes et le nombre de femmes qui étudient à la faculté.
- 4) Ecrire un algorithme qui donne l'âge moyen des étudiants de première année.
- 5) Ecrire un algorithme donnant l'étudiant le plus jeune, et son âge.
- 6) Ecrire un algorithme donnant l'étudiant le plus âgé, et son âge.
- 7) Ecrire un algorithme donnant le dernier étudiant inscrit à la faculté.

---

#### 4.6. ALGORITHME D'AJOUT

"Ajouter trois personnes dans le fichier "PERSONNEL" "

Précondition: Le fichier s'appelle "PERSONNEL",  
Un enregistrement est composé d'un nom et d'un âge.

Postcondition: Le fichier "PERSONNEL" doit contenir trois enregistrements de plus, le reste du fichier est inchangé.

Algorithme est le suivant:

Début:

ouvrir-en-écriture (“PERSONNEL”)  
répéter trois fois l’opération Inscrire-une-personne  
fermer (“PERSONNEL”)

Fin.

algorithme **Inscrire-une-personne**

Début :

demander son nom  
demander son âge  
écrire-dans (“PERSONNEL”)

Fin.

N.B.: On considère que l’opération demander est connue de l’ordinateur, et n’est terminée que si la réponse est reçue.

preuve: l’algorithme inscrire-une-personne est acceptable (en effet : il comporte trois instructions, et donc se termine ; il remplit le tampon avec le nom et l’âge, et l’écrit dans “PERSONNEL” si celui-ci est ouvert en écriture). L’algorithme principal est composé de deux instructions et d’une boucle contenant un algorithme acceptable, et qui est parcourue trois fois, donc se termine. L’ouverture est bien en écriture, et la fermeture est bien effectuée, l’algorithme est donc acceptable.

Trace:

Au début de l’algorithme, nous avons:

⊥	Paul 22	Georges 21	Nathalie 18	Sylvain 18	Robert 19	Valérie 21	⊥
---	------------	---------------	----------------	---------------	--------------	---------------	---

et le pointeur est indéfini.

ouvrir-en-écriture (“PERSONNEL”) provoque le positionnement du pointeur d’enregistrement sur la marque de fin de fichier, et l’association d’une zone tampon à ce fichier. Le tampon est indéfini.

⊥	Paul 22	Georges 21	Nathalie 18	Sylvain 18	Robert 19	Valérie 21	⊥
---	------------	---------------	----------------	---------------	--------------	---------------	---

↑  
pointeur  
d'enregistrement

--	--	--	--	--

indéfini indéfini
----------------------

tampon

demander nom et âge de la première personne à inscrire rempli le tampon:

⊥	Paul 22	Georges 21	Nathalie 18	Sylvain 18	Robert 19	Valérie 21	⊥
---	------------	---------------	----------------	---------------	--------------	---------------	---

↑  
pointeur  
d'enregistrement

--	--	--	--	--

Geny 20
------------

tampon

écrire-dans ("PERSONNEL") ajoute un enregistrement juste avant le pointeur:

⊥	Paul 22	Georges 21	Nathalie 18	Sylvain 18	Robert 19	Valérie 21	....
---	------------	---------------	----------------	---------------	--------------	---------------	------

....	Geny 20	⊥		
------	------------	---	--	--

↑  
pointeur  
d'enregistrement

Geny 20
------------

tampon

nouvelle demande de nom et âge:

⊥	Paul 22	Georges 21	Nathalie 18	Sylvain 18	Robert 19	Valérie 21	....
---	------------	---------------	----------------	---------------	--------------	---------------	------

....	Geny 20	⊥					
------	------------	---	--	--	--	--	--

↑  
pointeur  
d'enregistrement

Lydie 19
-------------

tampon

écrire-dans ("PERSONNEL") :

⊥	Paul 22	Georges 21	Nathalie 18	Sylvain 18	Robert 19	Valérie 21	....
---	------------	---------------	----------------	---------------	--------------	---------------	------

....	Geny 20	Lydie 19	⊥				
------	------------	-------------	---	--	--	--	--

↑  
pointeur  
d'enregistrement

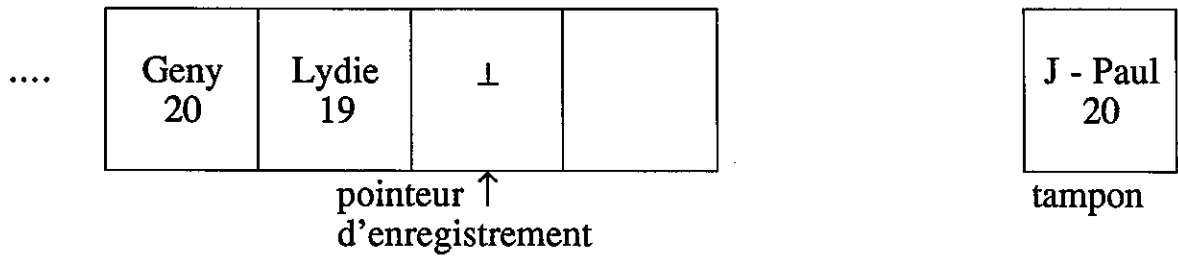
Lydie 19
-------------

tampon

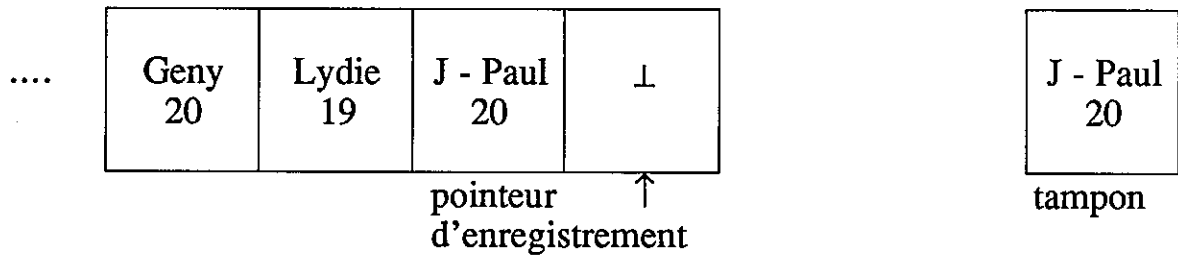
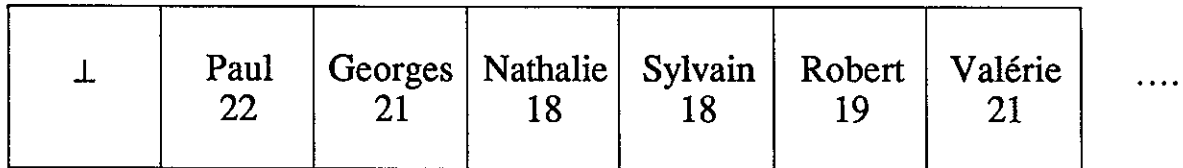
puis nouvelle demande de nom et âge:

⊥	Paul 22	Georges 21	Nathalie 18	Sylvain 18	Robert 19	Valérie 21	....
---	------------	---------------	----------------	---------------	--------------	---------------	------

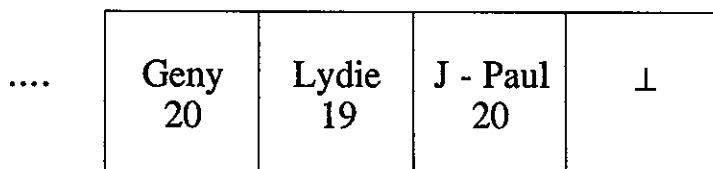
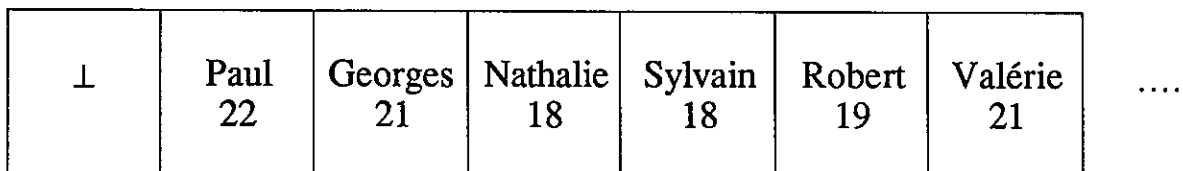




écrire-dans ("PERSONNEL"):



Enfin fermer ("PERSONNEL") supprime la relation tampon-fichier "PERSONNEL", ainsi que le pointeur.



Fin de l'exécution.

## 4.7. ALGORITHME DE CORRECTION

“Dans le fichier “PERSONNEL” il y a une erreur sur l’âge de Sylvain, on veut corriger cette erreur”.

Précondition: Le fichier s’appelle “PERSONNEL”,  
un enregistrement est composé d’un nom et d’un âge.  
Il existe un enregistrement dont le nom est Sylvain.

Postcondition: L’enregistrement dont le nom est Sylvain, doit avoir le bon âge.

Algorithme :

Les trois premières instructions recherchent l’enregistrement correspondant à Sylvain, et les trois dernières exécutent la modification.

Début :

Ouvrir-en-lecture (“PERSONNEL”)  
mettre le nom du tampon à vide.  
tant que nom ≠ “Sylvain” répéter lire (“PERSONNEL”)  
demander l’âge de Sylvain  
réécrire (“PERSONNEL”)  
fermer (“PERSONNEL”)

Fin.

NB: La mise à vide du nom du tampon permet d’éviter le cas fâcheux où la case mémoire réservée par l’ordinateur à ce nom du tampon contient déjà “Sylvain” avant toute lecture.

preuve :

La boucle tant que nom ≠ “Sylvain” répéter lire (“PERSONNEL”) permet de parcourir le fichier “PERSONNEL”, et s’arrête au bout d’un nombre fini d’itérations, car il existe un enregistrement dont le nom est Sylvain, (voir précondition). Réécrire, transfère le contenu du tampon dans le dernier enregistrement lu, qui est justement celui de Sylvain, l’algorithme est donc acceptable.

On pourrait généraliser cet algorithme en faisant demander le nom, et en vérifiant que ce nom existe:

Début:

ouvrir-en-lecture ("PERSONNEL")  
demander le nom à rechercher et le mettre dans nom-cherché  
mettre à vide le nom du tampon  
tant que (nom du tampon ≠ nom-cherché et non fin-de-fichier ("PERSONNEL")),  
    répéter lire ("PERSONNEL")  
si nom du tampon = nom-cherché alors faire l'opération suivante :  
    début  
        écrire-écran(âge)  
        demander le bon âge  
        réécrire ("PERSONNEL")  
    fin,  
sinon écrire-écran (nom-cherché, "n'est pas dans le fichier")  
fin-du-si.  
fermer ("PERSONNEL")

Fin.

Trace:

"PERSONNEL" en début d'algorithme:

⊥	Paul 22	Georges 21	Nathalie 18	Sylvain 18	Robert 19	Valérie 21	⊥
---	------------	---------------	----------------	---------------	--------------	---------------	---

Sylvain
---------

nom

ouvrir-en-lecture ("PERSONNEL") et mettre le nom du tampon à vide (le nom du tampon est la chaîne de caractères vide, mais l'âge du tampon est indéfini).

⊥	Paul 22	Georges 21	Nathalie 18	Sylvain 18	Robert 19	Valérie 21	⊥
---	------------	---------------	----------------	---------------	--------------	---------------	---

pointeur ↑  
d'enregistrement

Sylvain

nom

indéfini

tampon

La "troisième" instruction répète la lecture du fichier jusqu'à rencontrer l'enregistrement concernant Sylvain.

lire ("PERSONNEL"):

⊥	Paul 22	Georges 21	Nathalie 18	Sylvain 18	Robert 19	Valérie 21	⊥
---	------------	---------------	----------------	---------------	--------------	---------------	---

pointeur ↑  
d'enregistrement

Sylvain

nom

Paul  
22

tampon

lire ("PERSONNEL"):

⊥	Paul 22	Georges 21	Nathalie 18	Sylvain 18	Robert 19	Valérie 21	⊥
---	------------	---------------	----------------	---------------	--------------	---------------	---

pointeur ↑  
d'enregistrement

Sylvain

nom

Georges  
21

tampon

lire ("PERSONNEL"):

⊥	Paul 22	Georges 21	Nathalie 18	Sylvain 18	Robert 19	Valérie 21	⊥
---	------------	---------------	----------------	---------------	--------------	---------------	---

pointeur ↑  
d'enregistrement

Sylvain

nom

Nathalie  
18

tampon

lire ("PERSONNEL"):

⊥	Paul 22	Georges 21	Nathalie 18	Sylvain 18	Robert 19	Valérie 21	⊥
---	------------	---------------	----------------	---------------	--------------	---------------	---

pointeur ↑  
d'enregistrement

Sylvain

nom

Sylvain  
18

tampon

nom du tampon coïncide avec nom on passe donc à l'instruction suivante:  
demander l'âge de Sylvain: (qui ne modifie que le tampon)

⊥	Paul 22	Georges 21	Nathalie 18	Sylvain 18	Robert 19	Valérie 21	⊥
---	------------	---------------	----------------	---------------	--------------	---------------	---

pointeur ↑  
d'enregistrement

Sylvain

nom

Sylvain  
20

tampon

réécrire (“PERSONNEL”) met le contenu du tampon dans le dernier enregistrement lu:

⊥	Paul 22	Georges 21	Nathalie 18	Sylvain 20	Robert 19	Valérie 21	⊥
---	------------	---------------	----------------	---------------	--------------	---------------	---

pointeur ↑  
d'enregistrement

Sylvain
---------

nom

Sylvain 20
---------------

tampon

enfin fermer (“PERSONNEL”) supprime pointeur et tampon:

⊥	Paul 22	Georges 21	Nathalie 18	Sylvain 20	Robert 19	Valérie 21	⊥
---	------------	---------------	----------------	---------------	--------------	---------------	---

Sylvain
---------

nom

Fin de l'exécution.

#### 4.8. EXERCICES

On considère le fichier “ETUDIANT” de la faculté. Il y a un enregistrement par étudiant. Chaque enregistrement contient le nom, le sexe, l'âge, l'année d'étude, le lieu d'étude (AIX ou MARSEILLE), la division (A ou B), et le numéro du groupe de TD.

1) Ecrire un algorithme qui donne la liste des étudiants de première année MARSEILLE.

2) Ecrire un algorithme qui répartit les étudiants de première année de Marseille à raison d'un étudiant sur deux dans la division A, les autres étudiants de première année de Marseille allant dans la division B, et qui enregistre ce résultat dans le fichier.

- 3) Ecrire un algorithme qui répartit les étudiants de première année de la division A de Marseille en 5 groupes de TD notés M1, M2, M3, M4 et M5, et l'enregistre dans le fichier.
- 4) Ecrire un algorithme qui répartit les étudiants de première année de la division B de Marseille en 5 groupes de TD notés M6, M7, M8, M9 et M10, et l'enregistre dans le fichier.
- 5) Réaliser 3) et 4) en un seul algorithme.
- 6) Réaliser 2) et 5) en un seul algorithme.
- 7) Ecrire un algorithme qui recherche un étudiant, et l'inscrit s'il ne se trouve pas dans le fichier, sinon signale qu'il est déjà inscrit.
- 8) Ecrire un algorithme qui copie un fichier dans un autre.
- 9) Ecrire un algorithme qui concatène deux fichiers;

---

#### 4.9. ALGORITHME DE SUPPRESSION

“On désire supprimer du fichier “PERSONNEL”, l'enregistrement concernant Nathalie”.

Pour réaliser cela, nous aurons besoin d'un fichier intermédiaire, que nous appellerons “RESULTAT”, qui est vide au début de l'algorithme, et qui, en fin d'algorithme, contiendra les mêmes enregistrements que “PERSONNEL”, sauf l'enregistrement concernant Nathalie.

Précondition: Le fichier s'appelle “PERSONNEL”,  
un enregistrement est composé d'un nom et d'un âge.  
Il existe un enregistrement dont le nom est Nathalie.  
“RESULTAT” est un fichier vide, ayant la même structure  
que le fichier “PERSONNEL”.

Postcondition: Le fichier “RESULTAT” contient les mêmes enregistrements que “PERSONNEL”, sauf l'enregistrement concernant Nathalie, qui n'y figure pas.

## Algorithme:

### Début:

Ouvrir en lecture "PERSONNEL".

Ouvrir en écriture "RESULTAT".

Tant que non fin-de-fichier ("PERSONNEL") répéter l'opération suivante:

  début

    lire ("PERSONNEL")

    si (nom du tampon-PERSONNEL = "Nathalie"), alors on ne fait rien,  
    sinon on réalise l'opération suivante:

      début

        mettre tampon-PERSONNEL dans tampon-RESULTAT,

        écrire-dans ("RESULTAT")

      fin

    fin-du-si.

  fin

  fermer ("RESULTAT")

  fermer ("PERSONNEL")

### Fin.

Le fichier "RESULTAT" contenant ce qu'on désire, il ne reste plus qu'à détruire le fichier "PERSONNEL" qui ne sert plus, et à changer le nom du fichier "RESULTAT" qui s'appellera désormais "PERSONNEL". Le fichier "RESULTAT" n'est donc là que pour un usage temporaire, voilà pourquoi nous l'appellerons: **fichier temporaire**.

Nous ajouterons les opérations suivantes connues de l'ordinateur :

**Créer un fichier**, avec une structure donnée, qui crée un fichier séquentiel vide.

**Détruire un fichier**, qui supprime le fichier nommé.

**Renommer le fichier "A" en "B"** ; le fichier qui portait jusque là le nom "A", s'appelle désormais "B". Pour cela, aucun fichier de nom "B" ne doit exister avant l'application de cette instruction.

Nous obtenons alors la formulation suivante:

Précondition:       Le fichier s'appelle "PERSONNEL",  
                          un enregistrement est composé d'un nom et d'un âge.  
                          Il existe un enregistrement dont le nom est Nathalie.

Postcondition: L'enregistrement concernant Nathalie, ne figure plus dans le fichier "PERSONNEL" qui n'a pas été modifié par ailleurs.



Nous aurons besoin d'un fichier temporaire de même structure que "PERSONNEL", que nous appellerons "RESULTAT".

Algorithme:

Début :

Créer le fichier "RESULTAT", même structure que "PERSONNEL".

Ouvrir en lecture "PERSONNEL".

Ouvrir en écriture "RESULTAT".

Tant que non fin-de-fichier ("PERSONNEL") répéter l'opération suivante :

  début

    lire ("PERSONNEL")

    si (nom du tampon-PERSONNEL = "Nathalie"), alors on ne fait rien,  
    sinon on réalise l'opération suivante :

      début

        mettre tampon-PERSONNEL dans tampon-RESULTAT,  
        écrire-dans ("RESULTAT")

      fin

    fin-du-si.

  fin.

Fermer ("RESULTAT").

Fermer ("PERSONNEL").

Détruire ("PERSONNEL").

Renommer "RESULTAT" en "PERSONNEL".

Fin.

preuve:

L'ouverture du fichier "PERSONNEL" en lecture, positionne le pointeur de ses enregistrements sur le premier enregistrement de ce fichier, chaque passage dans la boucle fait lire "PERSONNEL", et, par définition de lire un **enregistrement**, fait progresser ce pointeur d'un enregistrement. Par définition, un fichier est fini, donc après un nombre fini de boucles, la fin du fichier sera rencontrée. Ainsi, après un nombre fini d'instructions, l'algorithme se terminera.

Chaque passage dans la boucle recopie un enregistrement de "PERSONNEL" dans "RESULTAT", sauf s'il s'agit de celui concernant Nathalie. Tous les enregistrements de "PERSONNEL" sont lus dans une boucle, d'où "RESULTAT" contient bien tous les enregistrements de "PERSONNEL" sauf celui concernant Nathalie, et rien que ceux-là, à la fin de la dernière boucle. D'où l'algorithme est acceptable.

#### 4.10. ALGORITHME D'AJOUT DANS UN FICHER TRIÉ

“Ajouter un enregistrement dans le fichier “PERSONNEL” qui est trié par nom dans l’ordre alphabétique ”.

Précondition : Le fichier s’appelle “PERSONNEL”,  
un enregistrement est composé d’un nom et d’un âge.  
Il est trié par ordre croissant alphabétique des noms.  
On connaît le nouveau-nom, et le nouvel-âge.

Postcondition: Le fichier “PERSONNEL” qui n’a pas été modifié par ailleurs, s’est enrichi d’un enregistrement qui contient le nouveau-nom et l’âge correspondant.

Le fichier “PERSONNEL” est trié par ordre croissant alphabétique des noms.

Nous aurons besoin d’un fichier temporaire de même structure que “PERSONNEL”, et que nous appellerons “RESULTAT”.

Algorithme:

Début:

Créer le fichier “RESULTAT”, même structure que “PERSONNEL”.  
Ouvrir en lecture “PERSONNEL”.  
Ouvrir en écriture “RESULTAT”.  
Lire (“PERSONNEL”).  
Tant que  
    (nom du tampon-PERSONNEL  $\leq$  nouveau-nom  
        et non fin-de-fichier (“PERSONNEL”)).

    Faire l’opération de transfert

Si (nom du tampon-PERSONNEL  $\leq$  nouveau-nom) réaliser:

    début

        mettre tampon “PERSONNEL” dans tampon “RESULTAT”,  
        écrire-dans (“RESULTAT”).

    fin.

Mettre nouveau-nom dans nom du tampon-RESULTAT.

Mettre nouvel-âge dans âge du tampon-RESULTAT.

Ecrire-dans (“RESULTAT”).

Tant que non fin-de-fichier (“PERSONNEL”).

    Faire l’opération de transfert

Si nom du tampon-PERSONNEL > nouveau-nom, alors faire  
début

mettre tampon-PERSONNEL dans tampon-RESULTAT,  
écrire-dans ("RESULTAT")

fin.

Fermer ("RESULTAT").

Fermer ("PERSONNEL").

Détruire ("PERSONNEL").

Renommer "RESULTAT" en "PERSONNEL".

Fin.

### Algorithme opération de transfert:

#### Début:

Mettre tampon-PERSONNEL dans tampon-RESULTAT.

Ecrire-dans ("RESULTAT").

Lire ("PERSONNEL").

Fin.

#### preuve:

L'algorithme opération de transfert se termine bien. Les deux boucles ne contiennent que des appels à l'opération de transfert, donc exécutent une lecture du fichier "PERSONNEL", et par suite ne seront parcourue qu'un nombre fini de fois. Et donc l'algorithme se termine.

Considérons l'invariant " "RESULTAT" est trié par ordre croissant des noms, et contient "PERSONNEL". Aucun enregistrement de "PERSONNEL"<sup>+</sup> n'est dans "RESULTAT"."

L'invariant est vérifié, après les trois premières instructions, juste avant la première lecture ("RESULTAT" étant vide, et "PERSONNEL" aussi. La première lecture fait perdre l'invariant, car le contenu du tampon-PERSONNEL n'est pas écrit dans "RESULTAT". Chaque passage dans la boucle de la cinquième instruction, rétablit l'invariant à la deuxième instruction de l'opération de transfert, mais la nouvelle lecture rend l'invariant faux. A la sortie de chaque application de l'opération de transfert, on se retrouve dans la même situation, à savoir qu'il faudrait écrire le contenu du tampon-PERSONNEL dans le fichier "RESULTAT" pour que l'invariant soit vérifié.

Deux cas se présentent alors:

**Soit** la fin du fichier a été rencontrée, et tous les noms figurant dans les enregistrements de "PERSONNEL" précèdent dans l'ordre alphabétique le nouveau-nom, alors la sixième instruction (l'instruction si), rétablit l'invariant, et les trois instructions suivantes ajoutent le nouvel enregistrement désiré, en conservant l'invariant, les deux instructions suivantes (la répétition, et le si) n'exécutant rien dans ce cas, on est donc rendu à la fermeture des fichiers, l'invariant étant vérifié.

**Soit**, la sixième instruction n'exécute rien, le tampon-PERSONNEL contient toujours un nom qui succède dans l'ordre alphabétique croissant le nouveau-nom, ce dernier donnant alors lieu à un nouvel enregistrement dans "RESULTAT", la dixième instruction (la répétition) n'est que le prolongement de la répétition de la cinquième instruction, mais alors, la onzième instruction (le si), rétablit l'invariant qui est donc, dans ce cas également vérifié à la fermeture des fichiers.

Or, justement à cette fermeture, "PERSONNEL" = "PERSONNEL", d'où "RESULTAT" contient bien "PERSONNEL". Il contient aussi le nouvel enregistrement, et est trié dans l'ordre alphabétique croissant des noms. Les deux dernières instructions permettent alors de satisfaire les postconditions.

Cet algorithme est bien acceptable.

---

#### 4.11. EXERCICES

- 1) Ecrire un algorithme qui vérifie si un fichier est trié, suivant un composant donné de la structure de l'enregistrement.
  - 2) Ecrire un algorithme qui recherche un élément, dans un fichier est trié, suivant un composant donné de la structure de l'enregistrement.
  - 3) Ecrire un algorithme qui permet de trier un fichier suivant un composant donné de la structure de l'enregistrement.
  - 4) Ecrire un algorithme qui supprime un enregistrement dans un fichier trié, suivant un composant donné de la structure de l'enregistrement.
-



*Animalcule Strange*

