



The slide has a decorative background featuring a globe icon on the left and a blue sun-like shape on the right, both with yellow rays. The word 'Objectifs' is centered in a large, purple, serif font. Below it is a bulleted list of objectives in blue text:

- Définir la structure enregistrement
- Définir les fichiers et les modes d'accès
- Mettre à profit les structures enregistrements et fichiers pour résoudre des problèmes

At the bottom left is the author's name 'DHAHRI Mbarek'. At the bottom center is the title 'Les enregistrements et les fichiers'. At the bottom right is the number '2'.

A – Les enregistrements

I - Introduction

Activité

- Une administration organise les informations relatives à ses employés dans une liste identique à la suivante :

Matricule	Nom & Prénom	Sexe	Nbre Enfants
M001	XXX Ahmed	M	0
M002	YYY Marwa	F	2
...

a – Donner la structure de données nécessaires pour les objets à utiliser.

b – Donner une déclaration algorithmique de ces objets.

DHAHRI Mbarek

Les enregistrements et les fichiers

3

I - Introduction

Réponse

- Cette liste comporte une information **alphanumérique** (Matricule), une information **numérique** (nombre d'enfants) et d'autres **alphabétiques** (nom & prénom, sexe)
 - La solution est d'utiliser 4 variables de type tableau

Objet	Type/Nature	Rôle
Mat	Tableau de 30 chaînes	
Nom	Tableau de 30 chaînes	
Sexe	Tableau de 30 caractères	
NbEnf	Tableau de 30 entiers	

DHAHRI Mbarek

Les enregistrements et les fichiers

4

I - Introduction

Remarques

- Toutes ces variable ne peuvent pas être regroupées dans un même tableau car un tableau ne peut contenir que des éléments de même type.
- Si nous voulons établir par exemple une structure comportant en même temps des informations alphanumérique, numérique et alphabétique, nous devons créer un nouveau type permettant de les regrouper. Il s'agit d'une nouvelle structure appelée **enregistrement** ou **article** (record en Pascal)

DHAHRI Mbarek

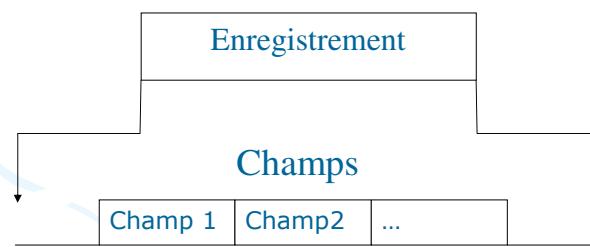
Les enregistrements et les fichiers

5

II – Définition et déclaration

a - Définition

- Un enregistrement est un type de données défini par l'utilisateur et qui permet de regrouper dans une même structure des données de différents types.



DHAHRI Mbarek

Les enregistrements et les fichiers

6

II – Définition et déclaration

b - Déclaration

En Algorithmique :

- Tableau de déclaration des nouveaux types (TDNT)

type
nom_type = enregistrement
champ1: type1
champ2: type2
....
champ_n : type_n
Fin nom_type

- Tableau de déclaration de objets (TDO)

Objet	Type/Nature	Rôle
Nom_Variable	Nom_type	rôle

DHAHRI Mbarek

Les enregistrements et les fichiers

7

II – Définition et déclaration

b - Déclaration

En Pascal :

```

type nom_type =record
    champ_1:type_1;
    champ_2:type_2;
    ...
    champ_n:type_n;
end;

var nom_variable : nom_type ;

```

DHAHRI Mbarek

Les enregistrements et les fichiers

8

II – Définition et déclaration

Activité

- Déclarer en algorithmique et en pascal une variable enregistrement représentant un élève (nom, prénom, date de naissance et sexe).
- Reprendre l'activité et déclarer le champ date de naissance comme étant un enregistrement.

Remarque

- Le champ d'une variable enregistrement peut être lui-même un enregistrement

DHAHRI Mbarek

Les enregistrements et les fichiers

9

III – Utilisation des enregistrements

a - Affectation

- Consiste à affecter des valeurs aux différents champs d'une variable enregistrement.

En Algorithmique	En Pascal
Nom_Variable.champ \leftarrow valeur	Nom_Variable.champ := valeur ;

DHAHRI Mbarek

Les enregistrements et les fichiers

10

III – Utilisation des enregistrements

a - Affectation

Exemple 1 :

```

PROGRAM portrait;
TYPE
  taille_t = 50 .. 250; { en cm }
  couleurs_t = (Blond, Brun, Roux, Bleu, Marron, Noir, Vert);
  cheveux_t = Blond .. Roux;
  yeux_t = Bleu .. Vert;
  personne_t = Record
    taille : taille_t;
    cheveux : cheveux_t;
    yeux : yeux_t;
  End;
  VAR bob, luc : personne_t;
  BEGIN
    bob.taille := 180;
    bob.cheveux := Brun;
    bob.yeux := Noir;
    luc := bob;
  END.
```

Il est possible d'affecter une variable enregistrement dans une autre à condition qu'ils aient la même structure.

DHAHRI Mbarek

Les enregistrements et les fichiers

11

III – Utilisation des enregistrements

a - Affectation

Exemple 2 :

```

type date = record
  jour_s : (lun, mar, mer, jeu, ven, sam, dim) ;
  jour : 1..31 ;
  mois : 1..12 ;
  annee : 0..99
end ;
type employe = record
  nom : string [30]
  prenom : string [20]
  masculin : boolean ;
  date_embauche : date ;
  nbenfants : 0..10
end ;
var e1, e2 : employe ;
• e1.date_embauche.annee désigne l'année d'embauche de l'enregistrement employe (cette information est de type 0..99),
• e2.date_embauche désigne la date d'embauche de l'enregistrement employe (il s'agit cette fois d'une information de type date)
```

DHAHRI Mbarek

Les enregistrements et les fichiers

12

III – Utilisation des enregistrements

b – Lecture

Au niveau de l'analyse	Au niveau de l'Algorithme	En Pascal
Nom_variable.champ=donnée	Lire (Nom_Variable.champ)	Readln (Nom_Variable.champ) ;

Activité:

- Écrire les instructions permettant de saisir à partir du clavier les champs de la variable enregistrement élève.

c – Écriture

Au niveau de l'analyse et de l'Algorithme	En Pascal
Ecrire (Nom_variable.champ)	Writeln (Nom_Variable.champ) ;

Activité:

- Écrire les instructions permettant d'afficher les champs de la variable enregistrement « élève ».

DHAHRI Mbarek

Les enregistrements et les fichiers

13

III – Utilisation des enregistrements

d – Structure Avec ... Faire

- L'instruction **Avec ... Faire** permet d'éviter l'utilisation répétée de la notation avec le point (variable.champ). Elle s'utilise avec une opération d'affectation , de lecture ou d'écriture .

Au niveau de l'analyse et de l'Algorithme	En Pascal
Avec (Nom_variable.champ) Faire { actions} Fin Avec	With Nom_variable do begin { actions} end;

Activité:

1. Réécrire les instructions des activités précédentes en utilisant la structure Avec ... Faire.

DHAHRI Mbarek

Les enregistrements et les fichiers

14

III – Utilisation des enregistrements

Application 1

- Établir une analyse modulaire, l'algorithme et la traduction en Pascal d'un programme qui permet de saisir et d'afficher un enregistrement nommé **LIVRE** caractérisé par : titre (chaîne de 30 c), nombre de pages (entier), nom de l'auteur (chaîne de 25 c), date d'édition (Enregistrement).

Application 2

- Établir une analyse modulaire, l'algorithme et la traduction en Pascal d'un programme qui créé des types intervalles **heure_t**, **minute_t** et **seconde_t**, puis un type enregistrement **temps_t**, puis d'afficher si deux temps t1 et t2 sont égaux.

DHAHRI Mbarek

Les enregistrements et les fichiers

15

III – Utilisation des enregistrements

e – Vecteur d'enregistrements

Il est possible de déclarer un tableau dont les éléments sont de type enregistrement. On définit tout d'abord le type enregistrement, puis on déclare le type tableau.

Exemple:

```

type
Personne = enregistrement
    nom : chaine[30]
    num_tel : chaine[10]
    num_cb : entier
fin personne

Tab = tableau de 100 personnes
  
```

DHAHRI Mbarek

Les enregistrements et les fichiers

16

III – Utilisation des enregistrements

Application 3

- Établir une analyse modulaire, l'algorithme et la traduction en Pascal du programme CLASSE qui permet de saisir les informations de N ($5 \leq N \leq 35$) élèves (Num, Nom, Prénom, Moyenne, Mention), puis d'afficher :
 - Tous les élèves
 - les élèves ayant une mention donnée et leur pourcentage par rapport au nombre total des élèves.

DHAHRI Mbarek

Les enregistrements et les fichiers

17

B – Les fichiers

I - Définition

Un **fichier** (*file*) est un ensemble structuré de données de même type stocké en général sur un support externe (disquette, disque dur, disque optique, flash disque, ...).

Il existe deux types de fichiers :

- Les **fichiers de données (typés)** (*File Of*) : sont des fichiers parfaitement formatés, c'est-à-dire qu'on peut y lire et écrire directement des variables de type structuré qui contiennent plusieurs champs de données.
- Les **fichiers textes** (*Text*) : sont écrits au format texte (chaînes de caractères, nombres)

DHAHRI Mbarek

Les enregistrements et les fichiers

18

II – Organisation des fichiers

L'organisation caractérise les modes d'implémentation des informations ou des enregistrements dans le fichier et fournit les propriétés d'accès.

- On distingue :

- L'organisation **séquentielle** : Les enregistrements sont stockés sur le support dans l'ordre dans lequel ils sont entrés.
- L'organisation **relative** : Un fichier en organisation relative se compose d'un certain nombre d'enregistrements de même taille et identifiés par un numéro d'ordre.

III – Types d'accès

Il s'agit des méthodes par lesquelles on lit ou on écrit un enregistrement d'un fichier.

On distingue :

- L'accès **séquentiel** : Pour lire une information particulière, il faut lire toutes les informations situées avant.
- L'accès **direct** : On peut accéder à l'information désirée grâce à son numéro d'ordre.

IV – Les fichiers de données

- Pour utiliser un ou des fichiers tout au long d'un programme, il faudra l'identifier par une variable dont le type est fonction de l'utilisation que l'on veut faire du fichier.
- En Algorithmique :

T.D.N.T

type
<code>nom_fichier = Fichier de type_composants</code>

T.D.O

Objet	Type/Nature	Rôle
Nom_logique	Nom_fichier	rôle

IV – Les fichiers de données

- En Pascal :

type

`nom_fichier = File Of type_composants ;`

var

`nom_logique : nom_fichier ;`

Activité:

- Déclarer en algorithmique et en Pascal la structure de fichier permettant de sauvegarder les moyennes des élèves.

IV – Les fichiers de données

1 – Traitement sur les fichiers

Étape 1 : Association (Assignation)

Pour un fichier, on trouve deux nom :

- Nom logique (interne) utilisé par le programme
- Nom physique (externe) utilisé par le système d'exploitation

Avant d'utiliser un fichier, il faut associer son nom logique au nom physique par la commande « **Associer** »

Syntaxe :

Au niveau de l'analyse et de l'Algorithme	En Pascal
Associer (Nom_logique, nom_physique)	Assign (Nom_logique, nom_physique) ;

1 – Traitement sur les fichiers

Exemple :

Au niveau de l'analyse et de l'Algorithme	En Pascal
Associer (fe, 'c:\eleve.dat')	Assign (fe, 'c:\eleve.dat');
Const NomFic = 'c:\4SI\donnees.dat' ; Assign (f , NomFic) ;	
Var NomFic : String ; Write ('sur quel fichier va-t-on travailler ? ') ; ReadIn (NomFic) ; Assign (F , NomFic) ;	
Const Repertoire = 'c:' ; Fichier = 'essai.dat' ; Assign (F , Repertoire + '\' + Fichier) ;	

1 – Traitement sur les fichiers

Étape 2 : Ouverture

L'instruction « **Recréer** » permet d'ouvrir un fichier et d'effacer son contenu s'il existe, sinon il sera créé.

Syntaxe :

Au niveau de l'analyse et de l'Algorithme	En Pascal
Recréer (Nom_logique)	Rewrite (Nom_logique) ;

Exemple :

Au niveau de l'analyse et de l'Algorithme	En Pascal
Recréer (fe)	Rewrite (fe) ;

1 – Traitement sur les fichiers

- L'instruction « **Ouvrir** » permet de renvoyer le pointeur au début du fichier pour pouvoir lire ou écrire à partir du début du fichier.

Syntaxe :

Au niveau de l'analyse et de l'Algorithme	En Pascal
Ouvrir (Nom_logique)	Reset (Nom_logique) ;

Exemple :

Au niveau de l'analyse et de l'Algorithme	En Pascal
Ouvrir (fe)	Reset (fe) ;

1 – Traitement sur les fichiers

Étape 3 : Écriture dans un fichier

L'instruction « **Ecrire** » permet d'écrire ou de modifier une valeur ou un enregistrement dans un fichier.

Syntaxe :

Au niveau de l'analyse et de l'Algorithmme	En Pascal
Ecrire (Nom_logique, variable)	Write (Nom_logique, variable) ;

Étape 4 : Lecture à partir d'un fichier

L'instruction « **Lire** » permet de lire une valeur ou un enregistrement à partir d'un fichier.

Syntaxe :

Au niveau de l'analyse et de l'Algorithmme	En Pascal
Lire (Nom_logique, variable)	Read (Nom_logique, variable) ;

1 – Traitement sur les fichiers

Étape 5 : Fermeture d'un fichier

L'instruction « **Fermer** » permet de fermer le ou les fichier(s) ouvert(s) à la fin du traitement.

Syntaxe :

Au niveau de l'analyse et de l'Algorithmme	En Pascal
Fermer (Nom_logique)	Close (Nom_logique) ;

Étape intermédiaire : test de fin du fichier

Au cours du traitement, nous pouvons tester si nous avons atteint la fin du fichier par l'instruction « **Fin_Fichier** ».

Syntaxe :

Au niveau de l'analyse et de l'Algorithmme	En Pascal
Fin_Fichier (Nom_logique)	Eof (Nom_logique) ;

IV – Les fichiers de données

- Exemple :

```
While Not Eof ( T ) Do
Begin
  { instructions de lecture d'une information}
  { instructions de traitement de cette information }
End ;
```

Directive de compilation

- La directive de compilation **I** permet de gérer les erreurs d'entrée-sortie : par défaut, elle est active et toutes les opérations d'entrée-sortie sont vérifiées par le système.
- Grâce à la fonction standard **IOresult**, il est possible de savoir si l'opération d'entrée-sortie s'est bien passée (retourne 0)

IV – Les fichiers de données

Directive de compilation

- Exemple : le programme demande le nom du fichier jusqu'à ce qu'un nom de fichier existant soit spécifié :

```
repeat
  write ('Donnez le nom physique du fichier ?');
  readIn (NomPhysique);
  assign (F, NomPhysique + '.DAT');
  {$I-}
  reset (F);
  {$I+}
  OK:=(IOresult = 0);
  if not OK then
    writeln ('impossible de trouver le fichier ', NomPhysique);
until OK;
```

IV – Les fichiers de données

Application 1

Écrire un programme qui permet de :

- enregistrer dans un fichier nommée « nombre.dat » des **entiers** composés de trois chiffres. La saisie se termine en répondant « O » à la question « Fin de la saisie (O/N) ? ».
- Afficher tous les nombres qui forment une suite croissante (ex : 123)

IV – Les fichiers de données

Application 2

- On veut informatiser la gestion d'une bibliothèque en sauvegardant dans un fichier nommée « Livres.dat » tous les livres. Chaque livre est caractérisé par :
 - Titre : chaîne (30)
 - Nom auteur : chaîne (30)
 - Date d'édition : chaîne (10)
 - Nombre de pages : entier

Écrire un programme qui permet de :

- Saisir et enregistrer les livres. La saisie se termine en répondant « N » à la question « Continuer (O/N) ? »
- Afficher tous les livres

2 – Les fichiers à accès direct

- Un fichier est dit à accès direct si on peut accéder directement à chacun de ses éléments. Cette opération se fait à l'aide de la procédure « **pointer** »

Syntaxe :

Au niveau de l'analyse et de l'Algorithmme	En Pascal
Pointer (Nom_logique, numéro)	Seek (Nom_logique, numéro) ;

NB : numéro désigne le numéro d'ordre de l'élément auquel on veut accéder. Il vaut **0** pour le 1^{er} élément.

- On peut connaître la taille d'un fichier (nombre d'éléments) grâce à la fonction « **Taille_Fichier** »

Syntaxe :

Au niveau de l'analyse et de l'Algorithmme	En Pascal
Taille_Fichier (Nom_logique)	FileSize (Nom_logique) ;

2 – Les fichiers à accès direct

Activité:

- Écrire les instructions qui permettent d'ajouter un livre de l'application précédente.

3 – Autres fonctions et procédures prédéfinies

- **Position_Fichier** : permet de connaître la position du pointeur dans un fichier.
- Syntaxe :

Au niveau de l'analyse et de l'Algorithmme	En Pascal
Position_Fichier (Nom_logique)	FilePos (Nom_logique) ;

- **Remarque** : Cette fonction ne peut pas s'appliquer sur un fichier **texte**.

Effacer : permet d'effacer le contenu d'un fichier (fermé ou ouvert).

- Syntaxe :

Au niveau de l'analyse et de l'Algorithmme	En Pascal
Effacer (Nom_logique)	Erase (Nom_logique) ;

Exemple :

```
...
Var F : File of char;
begin
  {$I-}
  Assign ( F , 'c:\essai.dat' ) ;
  Erase ( F ) ;
  If IOresult <> 0 Then
    Writeln ( 'impossible d''effacer le fichier' ) ;
  {$I+}
End.
```

DHAHRI Mbarek

Les enregistrements et les fichiers

35

Renommer : permet de changer le nom physique d'un fichier fermé.

Syntaxe :

Au niveau de l'analyse et de l'Algorithmme	En Pascal
Renommer (Nom_logique, Nouveau_nom_physique)	 Rename (Nom_logique , Nouveau_nom_physique) ;

- Exemple :**

```
Assign ( F , NouveauNom ) ;
{$I-}
Erase ( F ) ;
If IOresult <> 0 Then ; { si erreur ne rien faire }
Assign ( F , AncienNom ) ;
Rename ( F , NouveauNom ) ;
If IOresult <> 0 Then
  Writeln ( 'nouveau nom invalide ou ancien nom introuvable' ) ;
{$I+}
```

DHAHRI Mbarek

Les enregistrements et les fichiers

36

Tronquer : permet de tronquer un fichier, c'est-à-dire de supprimer tout ce qui se trouve après la position courante du pointeur.

- Syntaxe :

Au niveau de l'analyse et de l'Algorithmme	En Pascal
Tronquer (Nom_logique)	Truncate (Nom_logique) ;

Exemple 1 : détruit la dernière fiche d'un fichier

Seek (F , **FileSize** (F) - 1) ;

Truncate (F);

Exemple 2 : Celle-ci détruit l'ensemble du contenu d'un fichier. Elle est équivalent à **Rewrite** (F) :

Seek (F , 0) ;

Truncate (F);

DHAHRI Mbarek

Les enregistrements et les fichiers

37

Application 3

Écrire un programme permettant d'accéder d'une façon directe et de lire les données enregistrées dans le fichier « livre.dat » de l'application précédente. Ce programme permet de :

- Saisir le n° d'ordre de l'élément à lire
- Vérifier la validité de ce numéro
- Répéter les deux actions précédentes jusqu'à la réponse par « Non » à la question « Continuer (O/N) ? ».

DHAHRI Mbarek

Les enregistrements et les fichiers

38

Exercice 1

- 1) Écrire un programme qui remplit un fichier par N ($3 \leq N \leq 20$) entiers aléatoires compris entre 1 et 300.
- 2) Écrire un programme qui réalise les tâches suivantes :
 - 1 – Afficher tout le fichier
 - 2 – Ajouter un entier
 - 3 – Modifier un entier
 - 4 – Supprimer un entier
 - 5 – Insérer un entier

L'utilisateur doit choisir une actions parmi les cinq

Indication :

- Les actions ajouter, modifier, supprimer et insérer se feront à l'aide du numéro d'ordre de l'entier dans le fichier.

Exercice 2

Écrire un programme qui recherche un auteur donnée par dichotomie dans un fichier de livres (On suppose que les livres sont triés par auteur).

Indication :

- la fonction **RechDicho** renvoie le numéro de l'enregistrement vérifiant le critère de recherche, ou -1 si aucun enregistrement ne vérifie le critère
- G et D sont les indices courants (gauche et droite) de la recherche dichotomique dans le fichier.

Programme de recherche dichotomique

```

function RechDicho (var f: TypeFichier; v: t_livre) : integer;
var G, D, m : integer;
    trouve : boolean;
    courant : t_livre;
Begin
    G := 0; D := filesize (f) - 1; (* bornes de l'ensemble considéré *)
    trouve := false;
    while (G <= D and not trouve) do
        begin
            m := (G + D) div 2; (* calcul de la valeur du milieu *)
            seek (f, m); (* positionnement du repère de fichier *)
            read (f, courant); (* lecture de l'enregistrement *)
            if courant.auteur = v then trouve := true (* condition de sortie *)
            else if courant.auteur > v then D := m-1 (* trop à droite *)
            else G := m+1; (* trop à gauche *)
        end; (* while *)
        if trouve then RechDicho := m (* détermine l'état de sortie du prg *)
        else RechDicho := -1;
    end; (* RechDico *)

```

DHAHRI Mbarek

Les enregistrements et les fichiers

41

V – Les fichiers textes

1 – Définition

Les **fichiers textes** sont écrits au format texte (chaînes de caractères, nombres) dans lesquels on peut écrire et lire ligne par ligne ou à la file.

- Les caractères contenus dans un fichier de texte sont organisés en lignes, chacune terminée par une **marque de fin de ligne**. Après la dernière ligne, le fichier se termine par une **marque de fin de fichier**.

2 – Déclaration

En Algorithmique :

T.D.O

Objet	Type/Nature	Rôle
Nom_logique	Texte	rôle

En Pascal :

VAR nom_logique : **Text** ;

DHAHRI Mbarek

Les enregistrements et les fichiers

42

V – Les fichiers textes

Activité

- Déclarer un fichier texte nommée « FT » l'associer au nom physique « c:\4SI\essai.txt ».

3 – Fonctions et procédures prédéfinies

Les fonctions et procédures déjà vues pour les fichiers typés restent valable pour les fichiers textes, sauf : **FilePos**, **FileSize** et **Seek**.

- Fin_Ligne** : Fonction qui retourne « Vrai » quand le pointeur est positionnée en fin de ligne ou quand EOF est « Vrai ».

Syntaxe :

Au niveau de l'analyse et de l'Algorithmme	En Pascal
Fin_Ligne (Nom_logique)	EoLn (Nom_logique) ;

DHAHRI Mbarek

Les enregistrements et les fichiers

43

V – Les fichiers textes

- Ajouter** : Procédure qui ouvre le fichier et positionne son pointeur à la fin de ce dernier en vue d'ajouter des éléments.

Syntaxe :

Au niveau de l'analyse et de l'Algorithmme	En Pascal
Ajouter (Nom_logique)	Append (Nom_logique) ;

Exemple :

```

{$I-}
Append ( T ) ;
If IOresult <> 0 Then Rewrite ( T ) ; { si échec alors création }
{$I+}
  
```

DHAHRI Mbarek

Les enregistrements et les fichiers

44

V – Les fichiers textes

Exemple 2:

Lire un fichier contient les informations suivantes :
 Un prénom sur les 20 premières colonnes de chaque ligne
 Un nom sur les 20 colonnes suivantes

```
Var Ligne; Prenom, Nom : String ;
      p : word;
While Not Eof ( T ) Do
Begin
  ReadIn ( T , Ligne ) ;
  P := Pos ('` , Ligne ) ;
  Prenom := Copy ( Ligne , 1 , P - 1 ) ;
  Delete ( Ligne , 1 , P ) ;
  Nom := Ligne ;
End ;
```

Une autre solution serait de lire le fichier caractère par caractère et de reconstituer les chaînes nom et prénom en utilisant le premier espace comme indicateur de séparation entre les deux.

V – Les fichiers textes

Version lecture caractère par caractère	Version lecture ligne par ligne
<pre>while not eof (T) do begin while not eoln (T) do begin read (T, caractere); ... (* traiter le caractère lu *) end; readIn (T); End;</pre>	<pre>while not eof (T) do begin readIn (T, chaine_caracteres); ... (* traiter la chaîne lue *) End;</pre>

V – Les fichiers textes

Application 1

- 1) Écrire un programme qui affiche le nombre de lignes d'un fichier texte.**
- 2) Écrire un programme qui affiche le nombre de mots d'un fichier texte.**
- 3) Écrire un programme qui affiche le nombre de chiffres et de lettres contenu dans un fichier texte.**

DHAHRI Mbarek

Les enregistrements et les fichiers

49

ANNEXE

- **Tableau des correspondances entre procédures et types de fichiers**

Syntaxe	Type de fichiers associés	
	Lecture	Écriture
Reset (<i>f</i>)	-Text -File of	-File of
Rewrite (<i>f</i>)		-Text -File of
Append (<i>f</i>)		-Text

DHAHRI Mbarek

Les enregistrements et les fichiers

50

