

# LES ENREGISTREMENTS ET LES FICHIERS

Disponible sur [www.elbahi.jimdo.com](http://www.elbahi.jimdo.com)

## I- Les Enregistrements :

### 1- Introduction :

Dans votre lycée, chaque classe est organisée sous la forme d'une liste comme la suivante :

N°	Code	Nom & Prénom	Moyenne	Observations
1	C0120	Cherni Selim	14.25	Néant
2	K0235	Kefi Marwa	13.12	Redoublante
...	...	...	...	....
...	...	...	...	...
30	B3017	Bennour Raouf	11.75	Dispensé du sport

Le service scolarité de votre lycée vous demande de créer un programme informatique qui permet la saisie et le traitement de chaque classe sachant que le nombre maximal des élèves par classe ne dépasse pas 40 élèves.

#### Question :

Quels sont les objets (structures de données) à utiliser pour résoudre ce problème ?

#### Réponse :

#### Question :

Est ce qu'il est possible de regrouper toutes ces variables dans un même tableau ? (justifier)

#### Réponse :

Pour regrouper des variables de types différents (texte, numérique, ...) dans une même structure, il faut déclarer un nouveau type qui s'appelle ..... ou **ARTICLE** (**RECORD** en pascal)

## 2- Définition et déclaration :

Un enregistrement est un .....

.....

### ALGORITHMIQUE

TDNT

TYPES
<pre>Nom_type = Enregistrement     Champ1 : type1     Champ2 : type2     ...     Champn : typen Fin Nom_type</pre>

### PASCAL

TYPE

```
Nom_type = Record
    Champ1 : type1 ;
    Champ2 : type2 ;
    ...
    Champn : typen ;
End ;
```

TDO

objet	Type/nature	rôle
nom_objet	Nom_type	Enregistrement pour ...

### VAR

```
Nom_objet : Nom_type ;
```

### Activité01 :

- 1- Soit l'enregistrement **ELEVE** suivant formé de 5 champs, on vous demande de déclarer en algorithmique puis en pascal, l'enregistrement ELEVE

### ELEVE

Champ	Libellé	Type
N	Nom	Chaine [15]
P	Prénom	Chaine [20]
S	Section	Chaine
Ni	Niveau	Entier
G	Genre	Caractère

- 2- Déclarer l'élève **E**, en algorithmique puis en pascal

- 3- Donner quelques exemples d'objets pouvant avoir la structure d'un enregistrement

### 3- Utilisation :

#### 3.1 Affectation :

En algorithmique	En Pascal
<code>variable.champ ← valeur</code>	<code>variable.champ := valeur ;</code>

#### 3.2 Lecture :

Au niveau de l'analyse	Au niveau de l'algorithme	Au niveau du Pascal
<code>variable.champ = Donnée</code>	<code>Lire (variable.champ)</code>	<code>ReadLn (variable.champ);</code>

#### 3.3 Ecriture :

Au niveau de l'analyse et de l'algorithme	Au niveau du Pascal
<code>Ecrire (variable.champ)</code>	<code>Write (variable.champ) ;</code>

#### 3.4 La structure Avec ... Faire:

Au niveau de l'analyse et de l'algorithme	Au niveau du Pascal
<b>Avec variable Faire</b> {ensemble d'actions} <b>Fin Avec</b>	<b>With variable Do</b> <b>Begin</b> {ensemble d'actions} <b>End;</b>

#### Activité02 :

Soit la structure ELEVE suivante

ELEVE			
Champ	Libellé	Type	Contraintes
N	Nom	Chaine [15]	Non vide
P	Prénom	Chaine [20]	Non vide
MG	Moyenne générale	Réel	Entre 0 et 20
DN	Date de Naissance	Chaine	
G	Genre	Caractère	'H' ou 'F'

#### Travail A faire :

On vous demande de faire le programme pascal intitulé `gestion_eleve` qui permet de faire les tâches suivantes :

- 1- Remplir les champs d'un élève E en respectant les contraintes mentionnées dans le tableau.
- 2- Ajouter 1.5 à la moyenne générale de l'élève
- 3- Afficher toutes les informations de l'élève E.

**Remarque :**

- Il est impossible de lire ou d'écrire un enregistrement tout entier

**Exemple :**

Soit E un enregistrement, les instructions : **Ecrire(E) / lire (E)** sont (valides / invalides).

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

**Exemple :**

E1 et E2 deux enregistrements ayant la même structure, l'instruction **E1 ← E2** est (valide/ invalide)

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

**Exemple :**

Reprenez l'activité 2 et modifier le type du champ date de naissance (DN) par **date** au lieu de chaîne de caractères sachant que **date** est un enregistrement de 3 champs (j,m,a) de type entier chacun.

## II- Les vecteurs d'enregistrements :

Un vecteur d'enregistrements est .....

.....

**Exemple :**

Soit l'enregistrement **ELEVE** suivant formé de 4 champs :

Code	
Nom et prénom	
Genre	
Moyenne	

Le tableau T suivant est un vecteur formé par 4 enregistrements ELEVE.

*C'est le code de l'élève qui se trouve dans la case 3 du tableau T.* .....

T

<table border="1"> <tbody> <tr> <td>Code</td><td></td></tr> <tr> <td>Nom et prénom</td><td></td></tr> <tr> <td>Genre</td><td></td></tr> <tr> <td>Moyenne</td><td></td></tr> </tbody> </table>	Code		Nom et prénom		Genre		Moyenne		<table border="1"> <tbody> <tr> <td>Code</td><td></td></tr> <tr> <td>Nom et prénom</td><td></td></tr> <tr> <td>Genre</td><td></td></tr> <tr> <td>Moyenne</td><td></td></tr> </tbody> </table>	Code		Nom et prénom		Genre		Moyenne		<table border="1"> <tbody> <tr> <td>Code</td><td></td></tr> <tr> <td>Nom et prénom</td><td></td></tr> <tr> <td>Genre</td><td></td></tr> <tr> <td>Moyenne</td><td></td></tr> </tbody> </table>	Code		Nom et prénom		Genre		Moyenne		<table border="1"> <tbody> <tr> <td>Code</td><td></td></tr> <tr> <td>Nom et prénom</td><td></td></tr> <tr> <td>Genre</td><td></td></tr> <tr> <td>Moyenne</td><td></td></tr> </tbody> </table>	Code		Nom et prénom		Genre		Moyenne	
Code																																			
Nom et prénom																																			
Genre																																			
Moyenne																																			
Code																																			
Nom et prénom																																			
Genre																																			
Moyenne																																			
Code																																			
Nom et prénom																																			
Genre																																			
Moyenne																																			
Code																																			
Nom et prénom																																			
Genre																																			
Moyenne																																			

**Activité03 :**

Soit la structure **ELEVE** constituée par les 4 champs suivants :

Champ	Libelle	Type	Caractéristiques
C	Code	mot	Positif, formé de 4 chiffres
NP	Nom et Prénom	chaine de 30 caractères	Non vide
G	Genre	Caractère	'H' pour Homme et 'F' pour femme
MG	Moyenne générale	réel	Compris entre 0 et 20

The screenshot shows the Turbo Pascal Help window with the title "Type entier (Integer)". It states that Turbo Pascal provides five predefined integer types: Shortint, Integer, Longint, Byte, and Word. Below is a table of these types:

Type	Domaine	Format
Shortint	-128..127	8 bits signés
Integer	-32768..32767	16 bits signés
Longint	-2147483648..2147483647	32 bits signés
Byte	0..255	8 bits non signés
Word	0..65535	16 bits non signés

At the bottom, it says: "Tous les types entiers sont des type ordinaux."

**Travail A faire :**

Faire le programme pascal qui permet de :

- 1- Remplir un tableau T par N élèves  
( avec  $9 < N < 40$  )
- 2- Afficher le nom et le prénom de chaque élève admis.
- 3- Calculer et afficher le pourcentage des garçons admis ainsi que des filles admises.

**III- Les Fichiers :****1- Introduction :****Question :**

Après avoir remplir un tableau d'enregistrements, si vous redémarrez l'ordinateur est ce que vous trouvez les informations que vous avez saisi ?

**Réponse :**

.....

Donc : un tableau, une matrice, une variable de type entier, ... ne peuvent pas garder les valeurs des données de façon ..... et pour résoudre ce problème, il faut utiliser une nouvelle structure :

.....

**2- Définition :**

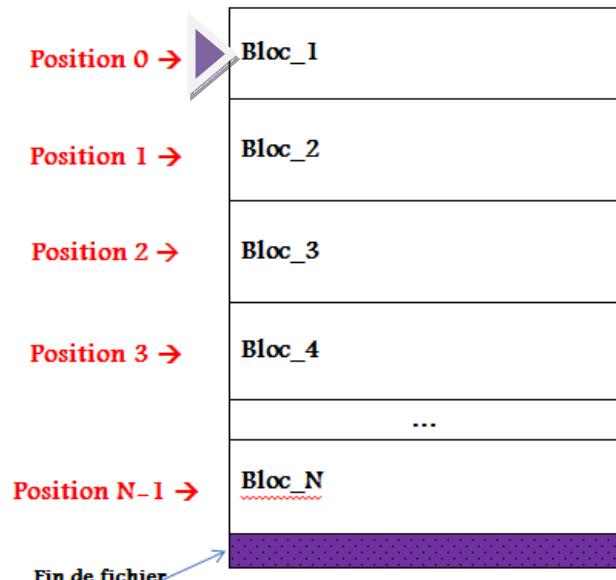
.....  
.....  
.....  
.....

**NB :** Un fichier peut contenir des caractères, des chaînes de caractères, des valeurs numériques, des enregistrements, ... et possède 2 noms : Un nom logique et un nom physique.

### Remarque :

### Comment peut-on représenter un fichier ?

- Pour accéder (pour lire ou modifier) à un bloc (un enregistrement), il faut le .....  
.....
- Le pointeur d'un fichier est un entier qui indique la position (l'adresse) dans le fichier à partir de laquelle la prochaine opération sera effectuée.
- Le premier bloc se trouve dans la position (adresse) zéro.
- Dans ce chapitre nous allons traiter 2 types de fichiers :
  - \* .....
  - \* .....



## 3- Fichier à accès direct :

### 3.1 Présentation et déclaration :

Un fichier à accès direct, si on peut accéder ..... à chacun de ses éléments.

#### En algorithmique

#### En pascal

##### TDNT

TYPE	
type_fichier = fichier de type_composants	

##### Type

type\_fichier = file of type\_composants ;

##### TDO

Objet	Type / nature	Rôle
Nom_logique	type_fichier	Fichier pour ...

##### Var

Nom\_logique : type\_fichier ;

### Remarque :

Un fichier possède 2 noms :

- ..... : le nom du fichier dans le programme (exemple : F)
- ..... : le nom du fichier sur le support de stockage (exemple : C:\bac2020\liste.dat)

**Activité04 :**

Nous voulons saisir et enregistrer les notes des élèves de votre classe dans un fichier de réels.

Faire les déclarations nécessaires (en algorithmique puis en pascal) pour réaliser ce travail.

Solution :

<b>TDNT</b> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center; padding: 5px;">TYPE</th></tr> </thead> <tbody> <tr> <td style="height: 50px; vertical-align: top; padding: 5px;"></td></tr> </tbody> </table>	TYPE		<b>Type</b>				
TYPE							
<b>TDO</b> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center; padding: 5px;">objet</th><th style="text-align: center; padding: 5px;">Type / nature</th><th style="text-align: center; padding: 5px;">Rôle</th></tr> </thead> <tbody> <tr> <td style="height: 50px; vertical-align: top; padding: 5px;"></td><td style="height: 50px; vertical-align: top; padding: 5px;"></td><td style="height: 50px; vertical-align: top; padding: 5px;"></td></tr> </tbody> </table>	objet	Type / nature	Rôle				<b>Var</b>
objet	Type / nature	Rôle					

Barrer ce qui est faux : F est le nom( logique / physique) du fichier.

**Activité05 :**

Maintenant, au lieu de saisir seulement les notes de vos camarades, on désire saisir toutes leurs données, sachant que les données qui nous intéressent sont les suivantes :

Fiche renseignement d'un élève :

Champ	Libellé	Caractéristiques
N	Numéro	Entier
NP	Nom et prénom	Chaîne [30]
NC1, NC2, NS	Les notes obtenues	Réel

Déclarer en algorithmique et en pascal les structures de données nécessaires.

### 3.2 Traitement sur les fichiers :

Rappelons que chaque fichier possède 2 noms : ..... et .....

Avant d'utiliser un fichier, il faut relier (associer) son nom logique à son nom physique grâce à la commande associer.

**L'association ou l'assignation :** (du nom logique d'un fichier à son nom physique)

Algorithmique	Pascal
Associer (nom logique, nom physique)	<code>ASSIGN (nom_logique, nom_physique) ;</code>

**Remarque :**

Une fois l'association faite, à chaque fois que nous voulons communiquer avec le fichier on utilise son nom logique.

**L'ouverture :**

Une fois on a fait l'association, on peut ouvrir le fichier soit pour lire, écrire ou réécrire.

+ Ouverture et création :

Algorithmique	Pascal
Recréer (nom logique)	<code>ReWrite (nom_logique) ;</code>

➔ Permet d'ouvrir un fichier et .....

+ Ouverture et remise à zéro :

Algorithmique	Pascal
Ouvrir (nom logique)	<code>ReSet (nom_logique) ;</code>

➔ Permet d'ouvrir un fichier et repositionner son pointeur à l'emplacement zéro.

**L'écriture dans un fichier :**

Algorithmique	Pascal
Ecrire (nom logique, variable)	<code>Write (nom_logique, variable) ;</code>

➔ Permet l'écriture ou la modification d'une valeur ou d'un enregistrement dans un fichier.

## Lecture à partir d'un fichier :

Algorithmique	Pascal
<b>Lire</b> (nom logique, variable)	<code>Read ( nom_logique, variable ) ;</code>

→ Permet la lecture d'une valeur ou d'un enregistrement à partir d'un fichier.

### Remarque

Après chaque écriture d'une valeur ou d'un enregistrement la valeur du pointeur .....

## Fermeture d'un fichier:

Algorithmique	Pascal
<b>Fermer</b> (nom logique)	<code>Close ( nom_logique ) ;</code>

→ A la fin du traitement, il faut fermer le fichier en utilisant la commande **Fermer** .

## Test de fin de fichier:

→ Au cours d'un programme, il est possible de tester si nous avons atteint la fin du fichier ou non en utilisant la fonction booléenne **Fin\_fichier (EOF)**.

Algorithmique	Pascal
<b>Fin_fichier</b> (nom logique)	<code>Eof ( nom_logique );</code>

## Activité06 :

Un entier est dit cubique ou d'Armstrong s'il est égal à la somme des cubes de ses chiffres comme 153.  
En fait 153 est cubique car  $153 = 1^3 + 5^3 + 3^3$

On désire faire un programme nommé **Nombres\_cubiques** qui remplit un fichier nommé logiquement **F** et physiquement par **D:\Nombres.dat** par N entiers positifs de 3 chiffres chacun, avec ( $4 < N \leq 20$ ).

Une fois rempli, le programme doit afficher chaque entier d'Armstrong qui se trouve dans le fichier suivi de son numéro d'ordre dans le fichier comme le montre l'exemple suivant:

Pour N = 5 et le fichier **Nombres.dat** suivant :

Le programme doit afficher :

Les entiers cubiques sont :

407 son ordre : 2

153 son ordre : 3

371 son ordre : 5

100  
407  
153  
900  
371

D:\Nombres.dat

## Travail à faire:

1- Faire la traduction pascal de votre programme.

**Activité07 : (exercice à la maison)**

Pour gérer ces produits, un magasin de vente de matériels informatique désire sauvegarder dans un fichier les fiches de renseignements de chacun de ses articles. Chaque produit se caractérise par les informations suivantes :

CHAMPS	TYPE	CARACTERISTIQUES
Code	octet	
Désignation	Chaine de caractères	Commence par un alphabet majuscule suivi de 3 chiffres
Prix d'achat	Réel	>0
Quantité en Stock	Réel	$\geq 0$

On désire faire le programme Pascal nommé **Gestion\_magasin** qui permet de :

- 1- saisir et sauvegarder sur le disque C : dans le fichier **MAGASIN.DAT** les fiches de renseignements des produits. la fin de la saisie sera confirmée si nous répondons par N (Non) à la question « **saisir un autre produit O/N ?** ».
- 2- Afficher la fiche de chaque produit (une fiche par page).
- 3- Calculer et afficher le nombre de produits ayant une quantité en stock nulle.

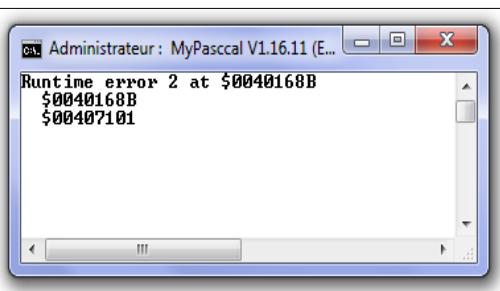
**Test d'existence d'un fichier :****Activité 08 :**

Sans créer le fichier « **C:\exercice.dat** », tester le programme pascal suivant :

```

1 Program EX;
2 uses wincrt;
3 type
4   fichier = file of char;
5
6 var
7   F:fichier;
8
9 begin
10 Assign (F , 'C:\exercice.dat' );
11 ReSet ( F );
12 end.

```

**Question :**

Pourquoi le programme affiche un message d'erreur ?

**Réponse :**

### Pour éviter ce problème :

Il est préférable de vérifier l'existence d'un fichier avant de l'utiliser.

A l'aide de directive `{$I-}`, l'utilisateur doit gérer lui-même ce type d'erreur grâce à la variable IOResult. (IOResult : Input Output Result).

### Comment faire ?

Ajouter ce qui se trouve dans les cadres au programme précédent

```

1  Program EX;
2  uses wincrt;
3  type
4    fiche = file of char;
5
6  var
7    F:fiche;
8
9  begin
10   Assign (F , 'C:\exercice.dat');
11   ReSet (F);
12   IF IOResult <> 0 then
13     write (' Fichier inexistant');
14 end.

```

### Positionner le pointeur du fichier:

Algorithmique	Pascal
<b>Pointer</b> (nom logique, numéro)	<code>Seek (nom_logique, numero);</code>

➔ Déplacer le pointeur à la position donnée

### Savoir le nombre d'éléments dans un fichier:

Algorithmique	Pascal
<b>Taille_fichier</b> (nom logique)	<code>FileSize (nom_logique);</code>

➔ Pour savoir le nombre d'éléments qui se trouvent dans le fichier

#### Remarque :

- Il ne faut pas positionner (seek) le pointeur après la fin du fichier.
- Les commandes seek et filepos ne sont utilisables qu'avec les fichiers à accès direct (de données)

**Activité 09 :**

On vous demande faire le programme pascal qui permet de :

- 1- Saisir un ensemble de caractères alphabétiques dans un fichier nommé physiquement « **C:\bac2020\alphabet.dat** » et logiquement « F » sachant que la saisie s'arrête une fois la réponse à la question « **Saisir autre caractère O/N ?** » est « N » ou « n » .
- 2- Afficher le nombre des voyelles et celui de consonnes.
- 3- Ajouter un module permettant d'ajouter un caractère alphabétique quelconque à une position donnée dans le fichier.

**3.3 Autres fonctions et procédures prédéfinies :****La fonction Position\_fichier :**

Algorithmique	Pascal
<b>Position_fichier</b> (nom logique)	<b>FilePos</b> (nom_logique);

→ Fournit la position de l'élément sur lequel se trouve le pointeur

**La procédure Effacer :**

Algorithmique	Pascal
<b>Effacer</b> (nom logique)	<b>Erase</b> (nom_logique);

→ Permet .....

**La procédure Renommer :**

Algorithmique	Pascal
<b>Renommer</b> (nom logique, nouveau nom physique)	<b>Rename</b> (nom_logique, nouveau_nom_physique);

→ Permet de renommer le nom physique d'un fichier

**Remarque :**

Avant de renommer un fichier, il faut le fermer.

**La procédure Tronquer :**

Algorithmique	Pascal
<b>Tronquer</b> (nom logique)	<b>Truncate</b> (nom_logique);

→ Permet de .....

## 4- Les fichiers texte :

### 4.1 Présentation et déclaration :

Un fichier texte (fichier ASCII) est un fichier à ..... formé uniquement par un ensemble de caractères.

Les fichiers textes constituent une catégorie particulière des fichiers car il contient des caractères de type « retour chariot » (CR), « fin de ligne » (EoLn) et « fin de texte » (CTRL-Z).

TDO

Objet	Type / nature	Rôle
Nom_logique	texte	Fichier pour ...

Var

Nom\_logique : text ;

### 4.2 Autres procédures et fonctions prédéfinies.

Autres les fonctions et procédures déjà vu, on peut utiliser d'autres fonctions et procédures de manipulation de chaîne de caractères.

#### La fonction Fin\_ligne (EoLn) :

Algorithmique	Pascal
Fin_ligne (nom logique)	EoLn (nom_logique);

➔ Fonction booléenne qui donne vrai si on se trouve sur le caractère CR (Retour chariot)

#### La procédure ajouter :

Algorithmique	Pascal
Ajouter (nom logique)	Append (nom_logique);

➔ Ouvre le fichier et positionne son pointeur à la fin du fichier. Seul l'ajout d'éléments est possible.

#### La procédure Lire\_nl :

Algorithmique	Pascal
Lire_nl (nom logique, variable )	ReadLn (nom_logique, variable);

➔ Permet la lecture de contenu d'une ligne, puis pointe la ligne suivante.

#### La procédure Ecrire\_nl :

Algorithmique	Pascal
Ecrire_nl (nom logique, valeur)	WriteLn (nom_logique, valeur);

➔ Permet l'écriture dans un fichier texte.