



## Tableaux , Matrices et Enregistrements

### I- Tableaux :

#### 1) Déclaration / Initialisation

Un tableau T est une variable structurée formée d'un nombre entier N de variables simples de même type, qui sont appelées les composantes du tableau. Le nombre de composantes N est alors la dimension du tableau.

Déclaration d'un type tableau :

**Nom\_tab : Tableau [premind . . deuxind] de type\_élément**

#### Exemples:

T1 : Tableau [1..50] d'entier

T2 : Tableau [1..20] de réel

T3 : Tableau [1..20] de caractère

**Remarque:** Il est également possible de définir un type tableau comme dans l'exemple suivant :

CONST

    nmax = 50

TYPE

    tab : Tableau [1..nmax] d'entier

VAR

    T : tab

#### 2) Traitements sur les tableaux :

##### a) Accès au éléments d'un tableau:

Considérons un tableau T de dimension N.

L'accès au premier élément du tableau se fait par T[1]

L'accès au dernier élément du tableau se fait par T[N]

##### b) Chargement d'un tableau :

Procédure CHARGEMENT ( VAR T : tab ; N :entier)

VAR

    i : entier

DÉBUT

    Pour i de 1 à N Faire

        Écrire ("T [" , i , "]" :")

        Lire (T[i])

    Fin pour



FIN

c) Affichage du contenu d'un tableau :

Procédure AFFICHE ( T : tab ; N : entier)

VAR

i : entier

DÉBUT

Pour i de 1 à N Faire

Écrire ( T[i] )

Fin Pour

FIN

## II- Matrices :

### 1) Déclaration / Initialisation :

Une matrice ou tableau à deux dimensions A et à interpréter comme un tableau unidimensionnel de dimension L dont chaque composante est un tableau unidimensionnel de dimension C.

On appelle L le nombre de lignes du tableau et C le nombre de colonnes du tableau.

Un tableau à deux dimensions contient L\*C composantes.

Déclaration d'un type matrice :

Nom\_tab : Tableau [premind. .deuxind , remind. .deuxind] de type\_élément

### Exemples :

M1 : Tableau [1..30, 1..30] d'entier

M2 : Tableau [1..20, 1..20] de réel

M3 : Tableau [1..20, 1..20] de caractère

### Remarque :

Il est également possible de définir une matrice comme dans l'exemple suivant :

CONST

NL = 30

NC = 20

TYPE

MAT : Tableau [1.. NL, 1.. NC] d'entier

VAR

M : MAT

### 2) Traitements sur les matrices :

#### a) Accès au composantes d'une matrice:

Considérons un tableau M de L lignes et C colonnes.

Les indices du tableau varient de 1 à L, respectivement de 1 à C.



La composante de la Nième ligne et Mième colonne est notée :  $A[N,M]$ .

b) Chargement d'une matrice:

Algorithme Chargement

VAR

M : Tableau [1.. 3, 1..4] d'entier

i , j : entier

DÉBUT

Pour i de 1 à 3 Faire

Pour j de 1 à 4 Faire

Écrire ("M [", i , ", " , j , "]" :")

Lire (M [i, j])

Fin pour

Fin pour

FIN

c) Affichage du contenu d'une matrice:

Algorithme Afficher

VAR

M : Tableau [1.. 3, 1..4] d'entier

i,j : entier

DÉBUT

Pour i de 1 à 3 Faire

Pour j de 1 à 4 Faire

Écrire ( M[i, j])

Fin pour

Fin pour

FIN

### III- Enregistrements :

1) Définition:

Contrairement aux tableaux qui sont des structures de données dont tous les éléments sont de même type, les enregistrements sont des structures de données dont les éléments peuvent être de différents types et qui se rapportent à la même entité sémantique. Les éléments qui composent un enregistrement sont appelés champs. Avant de déclarer une variable enregistrement, il faut avoir au préalable défini son type, c'est à dire le nom et le type des champs qui le composent. Le type d'un enregistrement est appelé type structuré.



## 2) Syntaxe:

Nom Enregistrement: Enregistrement  
  Nom Champ 1 : Type Champ 1  
  Nom Champ 2 : Type Champ 2  
  ...  
  Nom Champ n : Type Champ n  
Fin Enreg

### Exemples :

Personne: Enregistrement  
  Nom : Chaîne  
  Prénom : Chaîne  
  Age : Entier  
Fin Enreg  
Date: Enregistrement  
  Jour : Entier  
  Mois : Entier  
  Année : Entier  
Fin Enreg  
Point: Enregistrement  
  Num : Entier  
  Abscisse : Réel  
  Ordonnée : Réel  
Fin Enreg

Une fois qu'on a défini un type structuré, on peut déclarer des variables enregistrements exactement de la même façon que l'on déclare des variables d'un type primitif , suivant le modèle suivant : **Nom \_ Var : Nom \_ Enregistrement**

### Exemples :

P : Personne  
D1, D2 : Date  
A, B, C : Point

### Représentation:

Les enregistrements sont composées de plusieurs zones de données correspondant aux champs.



	Nom	Prénom	Age
P:	<input type="text"/>	<input type="text"/>	<input type="text"/>

	Jour	Mois	Année
D1:	<input type="text"/>	<input type="text"/>	<input type="text"/>

	Jour	Mois	Année
D2:	<input type="text"/>	<input type="text"/>	<input type="text"/>

	Num	Abscisse	Ordonné
A:	<input type="text"/>	<input type="text"/>	<input type="text"/>

	Num	Abscisse	Ordonné
B:	<input type="text"/>	<input type="text"/>	<input type="text"/>

	Num	Abscisse	Ordonné
C:	<input type="text"/>	<input type="text"/>	<input type="text"/>

### 3) Manipulation des enregistrements:

La manipulation d'un enregistrement se fait à travers ses champs. Comme pour les tableaux, il n'est pas possible de manipuler un enregistrement globalement, sauf pour affecter un enregistrement à un autre de même type (ou le passer en paramètre. Par exemple, pour afficher un enregistrement il faut afficher tous ses champs un par un.

### 4) Accès aux champs d'un enregistrement:

Alors que les éléments d'un tableau sont accédés par l'intermédiaire de leur indice, les champs d'un enregistrement sont accessibles à travers leur nom, grâce à l'opérateur '.'



**Nom\_Var.Nom\_Champ** représente la valeur mémorisée dans le champ de l'enregistrement Par exemple, pour accéder à l'âge de la variable P, on utilise l'expression : P.Age

**Attention** : le nom d'un champ est TOUJOURS précédé du nom de la variable déclarée avec le type enregistrement auquel il appartient. On ne peut pas trouver un nom de champ tout seul, sans indication de la variable.

Les champs d'un enregistrement, tout comme les éléments d'un tableau, sont des variables à qui on peut faire subir les mêmes opérations (affectation, saisie, affichage,... ). Exemple 1 : Saisie des données concernant deux personnes P1 et P2, puis affichage de la différence d'âge entre ces deux personnes.

easy ways