

# **Chapitre 4**

# **Structures de données de base**

# **en algorithmique**

# **Objectifs pédagogiques du chapitre**

## Objectif général

A la fin du chapitre, l'étudiant est capable d'utiliser les principales structures de données de base en algorithmique pour organiser les données.

# **Objectifs pédagogiques du chapitre**

## Objectifs spécifiques

A la fin du chapitre, l'étudiant est capable d'utiliser correctement une structure de données de type :

- tableau à une dimension
- tableau à deux dimensions
- enregistrement

# Contenu

- Introduction aux structures de données
- Introduction aux tableaux
- Tableaux à une dimension
- Tableaux à deux dimensions
- Enregistrements

# Introduction aux structures de données

- Une **structure de données** est une **organisation** ou une **structuration** des données ou encore un **type de données**.
- Les principales structures de données de base sont :
  - les types scalaires prédéfinis (entiers naturels, entiers relatifs, réels, booléens, caractères) ;
  - les types scalaires construits (chaîne de caractères, intervalle, ensemble) ;
  - les tableaux à une ou plusieurs dimensions ;
  - les enregistrements.

# Introduction aux structures de données

- ATTENTION !!!

**structures de contrôle ≠ structures de données**

Les 1<sup>res</sup> sont des instructions permettant d'orienter le fil du déroulement de l'algorithme tandis que les 2<sup>ndes</sup> sont une organisation ou une structuration des données.

# Introduction aux tableaux

- On appelle **tableau** une **structure de données** constituée de **plusieurs éléments de même type** qu'on peut référencer au moyen d'un indice.
- Un tableau correspond à une **suite/succession finie de variables/éléments de même type**, évitant à l'utilisateur l'emploi d'un grand nombre de variables pour lesquelles il lui aurait fallu trouver des noms distinctifs.

# Introduction aux tableaux

- Exemple : on voudrait conduire une étude sur la taille des étudiants de la licence TC1 de l'ESI.

# Introduction aux tableaux

Une façon rébarbative de le faire serait de déclarer une liste de 200 variables de type réel, chacune associée à la taille d'un étudiant :

**t1** est de type réel, correspondant à la taille du 1<sup>er</sup> étudiant ;

**t2** est de type réel, correspondant à la taille du 2<sup>e</sup> étudiant ;

**t3** est de type réel, correspondant à la taille du 3<sup>e</sup> étudiant ;

**t4** est de type réel, correspondant à la taille du 4<sup>e</sup> étudiant ;

**t5** est de type réel, correspondant à la taille du 5<sup>e</sup> étudiant ;

**t6** est de type réel, correspondant à la taille du 6<sup>e</sup> étudiant ;

...

**t199** est de type réel, correspondant à la taille du 199<sup>e</sup> étudiant ;

**t200** est de type réel, correspondant à la taille du 200<sup>e</sup> étudiant.

# Introduction aux tableaux



Quelles difficultés aurions-nous eues s'il y avait mille, dix mille, un million de variables à manipuler ?

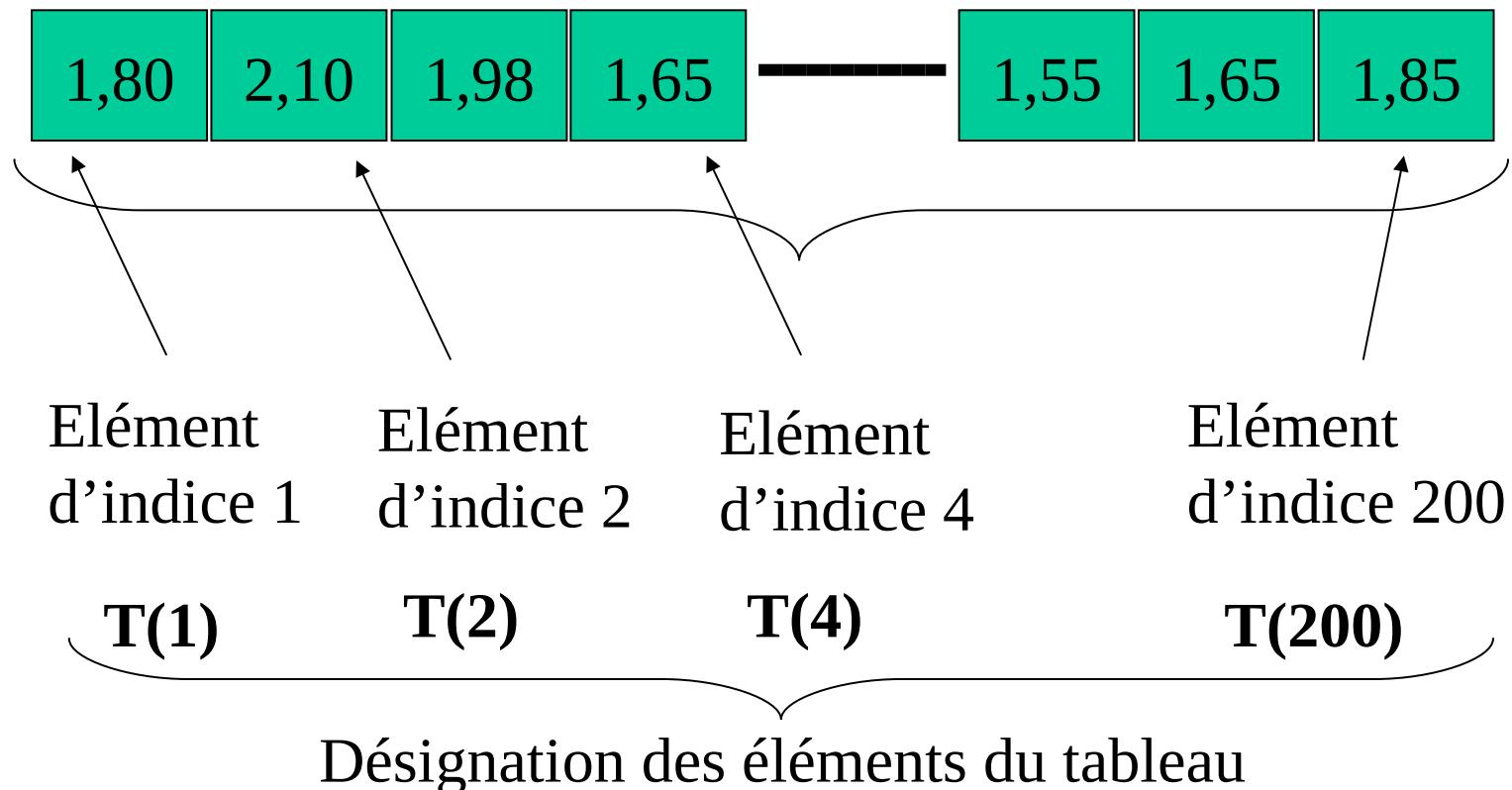
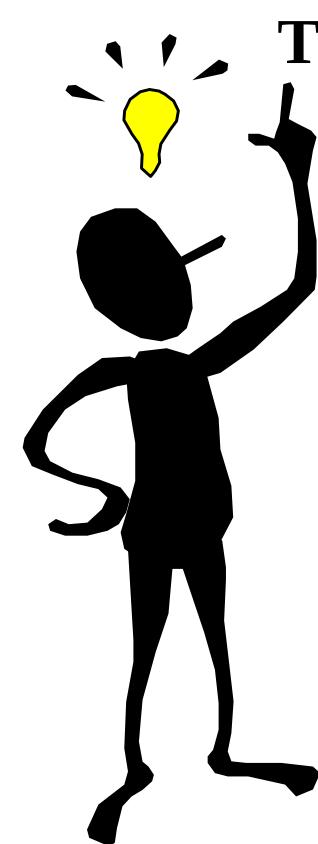
# Tableaux à une dimension

On les appelle aussi :

- *tableaux à une entrée*
- *vecteurs*

# Tableaux à une dimension

- Exemple



# Tableaux à une dimension

- Déclaration du type tableau



T est de type tableau de N objets



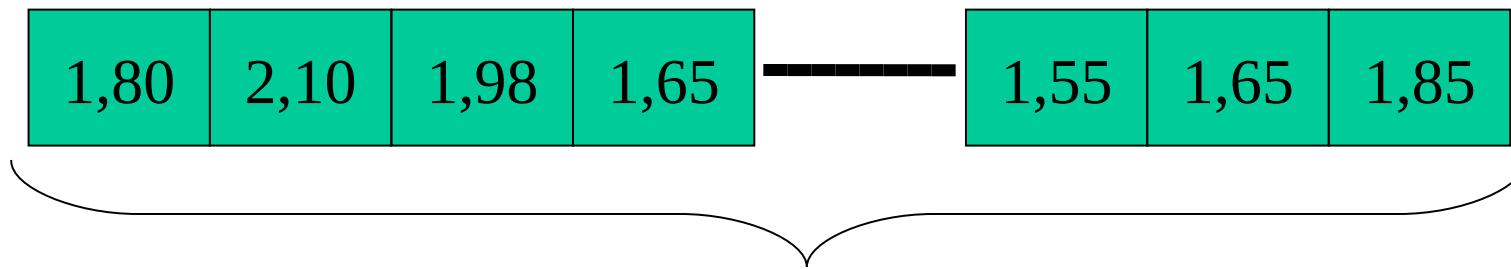
T : tableau [1..N] objets ← **Déclaration de type**

Ce qui traduit le fait que les objets de T soient  
indicés de 1 à N

Remarque : les indices peuvent démarrer n'importe  
quand pourvu qu'ils soient croissants

# Tableaux à une dimension

- Localisation des éléments / Encombrement du tableau



Elément d'indice 1 à l'adresse de début du tableau (@T)

Elément d'indice 2 à l'adresse de début du tableau + 1x (taille d'un élément)

Elément d'indice 3 à l'adresse de début du tableau + 2x (taille d'un élément)

Elément d'indice 4 à l'adresse de début du tableau + 3x (taille d'un élément)

Elément d'indice 200 à l'adresse de début du tableau + 199x (taille d'un élément)

1 bit : la valeur élémentaire de décomposition de l'information.

Elle vaut 0 ou 1 (chiffres de l'alphabet binaire)

Par comparaison, on a les chiffres décimaux (arabes) qui constituent l'alphabet décimal qui est : 0, 1, 2, 3, 4, 5, 6, 7, 8, 9.

Un paquet de 8 bits consécutifs représente un OCTET (BYTE).

Exemple :

00010100

10000011

1 Ko (kilo-octet) = 1000 octets

1 Kilo =  $2^{10}$  = 1024

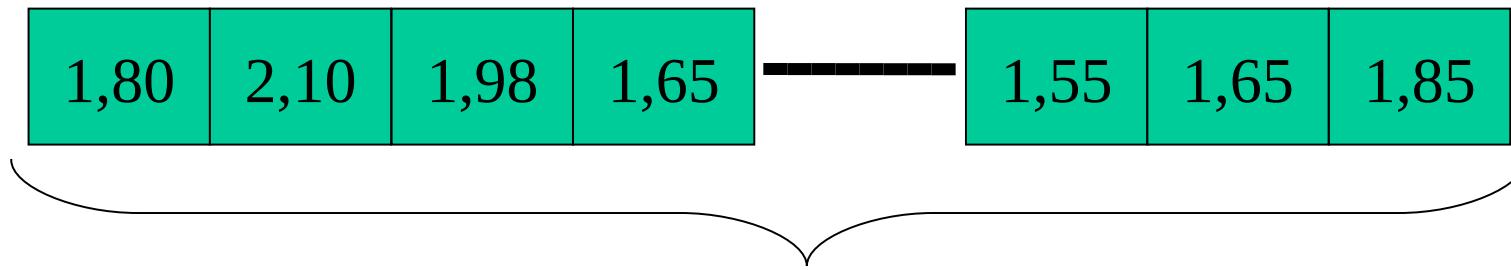
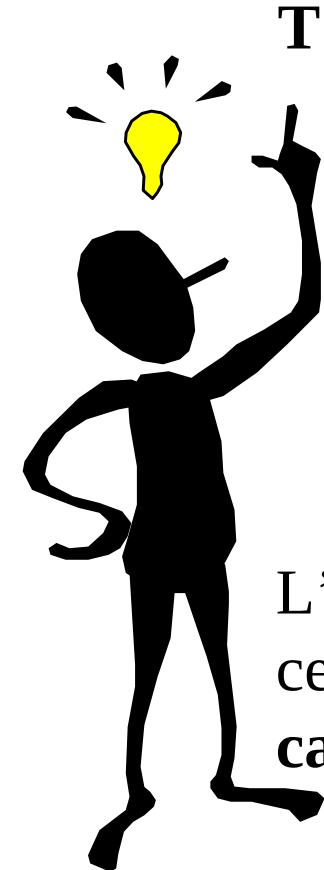
1 Kilo-binaire octets = 1 Kio = 1024 octets

1 Mega =  $10^6$  =

1 Mega-binaire =  $2^{20}$  => 1 Mio =  $2^{20}$  octets

# Tableaux à une dimension

- Localisation des éléments / Encombrement du tableau



A cause de la présence d'éléments de même type dans le tableau, l'ordinateur peut facilement **repérer chaque élément du tableau, pourvu que sa position relative** au début du tableau (indice ou rang) soit connue.

L'**encombrement du tableau**, i.e. l'espace mémoire centrale occupé par le tableau, peut être facilement calculé : **nombre d'éléments du tableau X taille d'un élément**.

# Tableaux à une dimension

- Opérations sur les tableaux à une dimension
  - Initialisation
  - Affichage
  - Repérage/recherche d'une valeur dans le tableau
  - Somme des éléments de deux tableaux
  - Calcul de la moyenne des éléments du tableau
  - Comptage des occurrences d'une valeur dans le tableau
  - Etc.

# Tableaux à une dimension

## Exercice 1

Écrire un algorithme permettant d'initialiser interactivement un tableau de 10 entiers naturels.

## Exercice 1bis

Écrire un algorithme permettant d'initialiser interactivement un tableau A donné de 10 entiers naturels.

## Exercice 2

Écrire un algorithme permettant d'afficher les 50 valeurs réelles d'un tableau. On suppose le tableau initialisé.

# Tableaux à une dimension

## Exercice 3

On suppose avoir deux tableaux A et B initialisés avec les valeurs correspondant aux notes de 200 étudiants pour les 2 évaluations en algorithmique. Ecrire un algorithme permettant de calculer et de conserver la moyenne des 2 évaluations de chacun des étudiants.

## Exercice 4

On suppose avoir un tableau T initialisé avec 1000 réels. Ecrire un algorithme permettant de récupérer la plus petite valeur de ce tableau et de l'afficher.

# Tableaux à une dimension

## Exercice 5

On suppose avoir un tableau T initialisé avec 1000 réels. Ecrire un algorithme permettant de récupérer la plus petite et la plus grande valeur de ce tableau.

## Exercice 6

On suppose avoir un tableau T initialisé avec 1000 réels. Ecrire un algorithme permettant de compter le nombre de valeurs réelles nulles de T.

# Tableaux à une dimension

## Exercice 7

On suppose avoir un tableau T initialisé avec 200 entiers relatifs. Ecrire un algorithme permettant de compter le nombre de valeurs strictement positives de T.

## Exercice 8

On suppose avoir un tableau T initialisé avec 20 entiers relatifs. Ecrire un algorithme permettant de retrouver l'indice du premier 0 du tableau quand celui-ci existe effectivement.

# Tableaux à une dimension

## Exercice 9

On suppose avoir un tableau T initialisé avec 10 entiers naturels. Ecrire un algorithme permettant de calculer le produits des valeurs de ce tableau.

## Exercice 10

On suppose avoir un tableau T initialisé avec N ( $N > 0$ ) valeurs réelles. Ecrire un algorithme permettant à l'utilisateur de saisir une valeur dont il faudra calculer le nombre d'apparitions dans le tableau T.

# Tableaux à deux dimensions

On les appelle aussi :

- *tableaux à deux entrées*
- *matrices*

# Tableaux à deux dimensions

- Définition : c'est un tableau de tableaux

Exemple : on veut traiter les notes de français, d'anglais et de maths de 4 étudiants dénommés A, B, C et D.

Solution n°1 : créer 4 tableaux de 3 réels.

A	6,50	12,75	11
B	16,50	10,75	10
C	9,50	9,75	11,50
D	13,50	12,50	14

# Tableaux à deux dimensions

Comment gérer les notes de la centaine d'étudiants ?



Quelles difficultés avec la centaine de noms de variables à manipuler ?

# Tableaux à deux dimensions

Solution n°2 : créer un tableau T de 4 tableaux de 3 réels qui sera déclaré comme suit :

T : tableau [1..4, 1..3] de réels.

T

6,50	12,75	11
16,50	10,75	10
9,50	9,75	11,50
13,50	12,50	14

# Tableaux à deux dimensions

T

6,50	12,75	11
16,50	10,75	10
9,50	9,75	11,50
13,50	12,50	14

- Ligne 1 (L1) associée aux notes de l'étudiant A
- Ligne 2 (L2) associée aux notes de l'étudiant B
- Ligne 3 (L3) associée aux notes de l'étudiant C
- Ligne 4 (L4) associée aux notes de l'étudiant D

Colonne 1  
pour les notes  
de français

Colonne 2  
pour les notes  
d'anglais

Colonne 3  
pour les notes  
de maths

# Tableaux à deux dimensions

13,50	12,75	11
16,50	10,75	10
9,50	9,75	11,50
13,50	12,50	14

Lignes {

Colonnes }

Désignation des éléments de la matrice

$T(i, j)$  avec :

i comme indice de la ligne  
j comme indice de la colonne

**Exemples :**

10 correspond à  $T(2, 3)$

11 correspond à  $T(1, 3)$

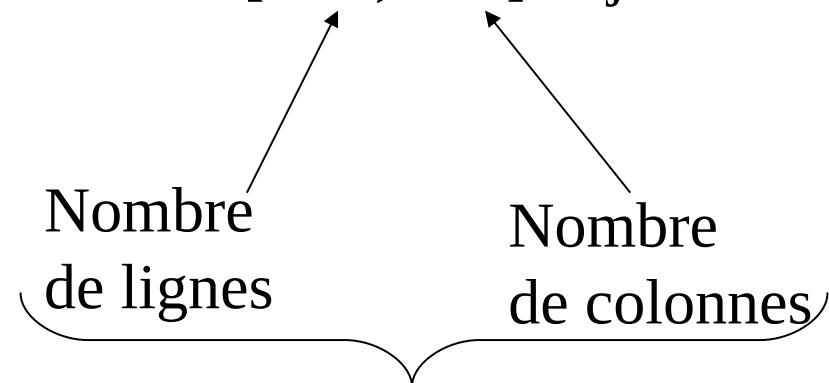
13,50 correspond à  $T(4, 1)$

# Tableaux à deux dimensions

## Généralisation

Une matrice **M** de **N** lignes et de **P** colonnes d'objets sera déclarée :

**M** : tableau de [1..N, 1..P] objets



**M**  $\Leftrightarrow$  Tableau de N tableaux [1..P] objets

# Tableaux à deux dimensions

## Représentation d'une matrice en mémoire centrale

En mémoire centrale, une matrice **M** de **N** lignes et de **P** colonnes d'**objets** est représentée à partir de l'adresse  **$\alpha$**  comme **un tableau à une dimension** constitué par la mise bout à bout des **N** lignes de **P** objets qui le constituent.

# Tableaux à deux dimensions

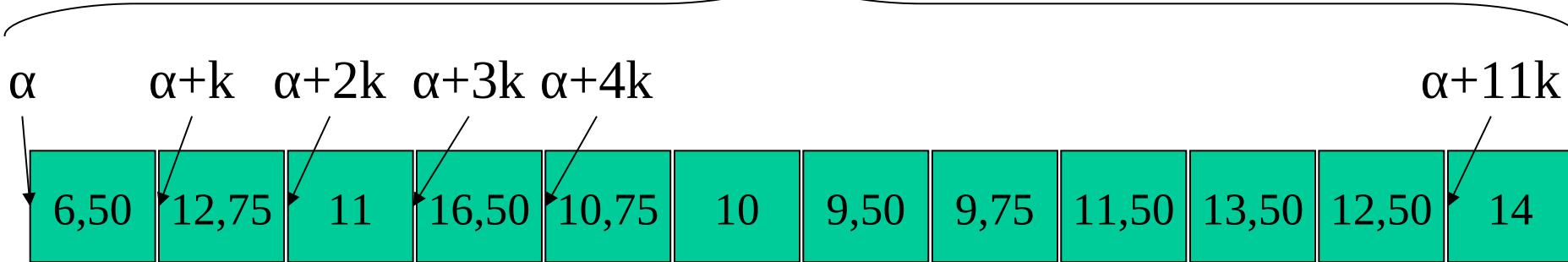
T

6,50	12,75	11
16,50	10,75	10
9,50	9,75	11,50
13,50	12,50	14

Si un élément de type **objet** de T est représenté sur k octets, alors en mémoire centrale on aura ce qui suit :



Adresses des éléments de T en mémoire centrale



T

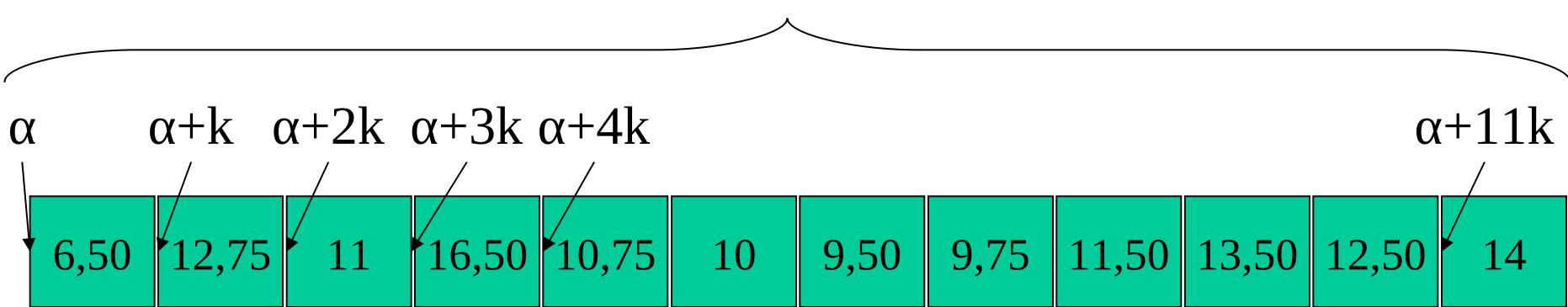
# Tableaux à deux dimensions

Soit  $\alpha$  l'adresse de début du tableau  $T[1..N, 1..P]$  **objet** en mémoire centrale.

Si un élément de type **objet** est représenté sur  $k$  octets, alors en mémoire centrale l'élément  $T(i, j)$  tel que  $1 \leq i \leq N$  et  $1 \leq j \leq P$  sera logé à l'adresse :

$$\alpha + (i-1)*P*k + (j-1)*k$$

Adresses des éléments de  $T$  en mémoire centrale



# Tableaux à deux dimensions

- Opérations sur les matrices
  - Initialisation d'une matrice
  - Affichage (ligne par ligne ou colonne par colonne) des valeurs d'une matrice
  - Repérage/recherche d'une valeur dans une matrice
  - Matrice Somme des éléments de deux autres matrices
  - Calcul de la moyenne des éléments d'une matrice
  - Comptage des occurrences d'une valeur dans la matrice
  - Etc.

# Tableaux à deux dimensions

## Exercice 1

Ecrire un algorithme permettant d'initialiser interactivement une matrice A de deux lignes et de trois colonnes d'entiers naturels.

## Exercice 1bis

Ecrire un algorithme permettant d'initialiser interactivement une matrice A de deux lignes et de cent colonnes d'entiers naturels.

## Exercice 1ter

Ecrire un algorithme permettant d'initialiser interactivement une matrice A de 1000 lignes et de 500 colonnes d'entiers naturels.

# Tableaux à deux dimensions

## Exercice 2

Ecrire un algorithme permettant d'afficher ligne par ligne, les 15 valeurs réelles d'une matrice A[1..5, 1..3]. On supposera la matrice déjà initialisée.

# Tableaux à deux dimensions

## Exercice 3

On suppose avoir deux matrices X et Y de L lignes et C colonnes de réels initialisées.

Ecrire un algorithme permettant de calculer la matrice Z contenant les moyennes des valeurs de X et Y, c'est-à-dire :

$$Z(i, j) = \text{moyenne}(X(i, j), Y(i, j)), i \in [1..L] \text{ et } j \in [1..C].$$

# Tableaux à deux dimensions

## Exercice 4

On suppose avoir une matrice  $M[1..N, 1..P]$  de réels initialisée.

Ecrire un algorithme permettant de récupérer la première plus petite valeur de cette matrice, de l'afficher ainsi que les indices de ligne et colonne lui correspondant. Le parcours de la matrice s'effectue de la 1<sup>re</sup> ligne à la dernière ligne.

# Tableaux à deux dimensions

## Exercice 5

On suppose avoir une matrice M de 3 lignes et 5 colonnes initialisée avec des valeurs réelles.

Ecrire un algorithme permettant de compter le nombre de valeurs strictement positives dans cette matrice.

# Tableaux à deux dimensions

## Exercice 6

On suppose avoir une matrice carrée M initialisée avec 25 entiers relatifs. Écrire un algorithme permettant de rechercher dans la matrice, une valeur V saisie par l'utilisateur. En cas de succès de la recherche, les coordonnées en ligne et en colonne de la valeur dans la matrice sont à afficher.

# Tableaux à deux dimensions

## Exercice 7

On suppose avoir une matrice d'entiers naturels M de L lignes et de C colonnes initialisée. Écrire un algorithme permettant à l'utilisateur de saisir une valeur entière dont il faudra calculer le nombre d'apparitions dans la matrice.

# Tableaux à deux dimensions

## Exercice 8

Les 12 valeurs réelles suivantes correspondent à une matrice M implantée en mémoire centrale.

6,51	10,52	-0,11	19,03	17,35	10,40	9,55	-2,95	11,50	13,50	-9,50	1,45
------	-------	-------	-------	-------	-------	------	-------	-------	-------	-------	------

- 1) Si la déclaration de M est  $M[1..6, 1..2]$  de réels,  
donnez les valeurs de  $M[2,1]$ ,  $M[4,1]$ ,  $M[6,2]$ ,  $M[6,2]$
  
- 2) Si la déclaration de M est  $M[1..4, 1..3]$  de réels,  
donnez les valeurs de  $M[2,1]$ ,  $M[4,1]$ ,  $M[3,3]$ ,  $M[4,2]$

# **Enregistrements**

On appelle **enregistrement** une **structure de données** correspondant à un regroupement de plusieurs éléments de différents types qu'on appelle **champs ou rubriques**.

# Enregistrements

## Exemple 1 : l'adresse d'une entreprise

l'adresse d'une entreprise commerciale est beaucoup employée dans les correspondances envoyées par l'entreprise. Elle est constituée par le nom de l'entreprise, le numéro de l'entrée dans la rue, le nom de la rue, le numéro de la boîte postale, le code postal de la ville, le nom de la ville et le nom du pays.

# Enregistrements

## Exemple 1 :

- le nom de l'entreprise      ← chaîne de 30 caractères
- le numéro de l'entrée dans la rue      ← Entier naturel
- le nom de la rue      ← chaîne de 40 caractères
- le numéro de la boîte postale      ← Entier naturel
- le code postal de la ville      ← Entier naturel
- le nom de la ville      ← chaîne de 35 caractères
- le nom du pays      ← chaîne de 20 caractères

Différents types

# Enregistrements

## Exemple 2 : la date

la date est une information que nous employons régulièrement. Elle est constituée d'un nom de jour, le numéro de jour (quantième), du nom du mois, le numéro de l'année.

# Enregistrements

## Exemple 2 : la date

- **le nom du jour** ← chaîne de 8 caractères max.
- **le numéro du jour dans le mois** ← Entier naturel compris entre 1 et 31
- **le nom du mois** ← chaîne de 9 caractères max.
- **le numéro de l'année** ← Entier naturel

# **Enregistrements**

## Déclaration du type enregistrement

**NomTypeEnregistrement = *Type Enregistrement regroupant***

*Liste déclarative  
de tous les champs,  
telle que donnée dans les  
paragraphes de déclaration  
des données en entrée et sortie*

***FinEnregistrement***

# Enregistrements

## Exemples de déclaration du type enregistrement

**DATE** = *Type Enregistrement regroupant*

**nomJour** est de type chaîne de 8 caractères, correspondant au nom du jour ;

**numeroJour** est de type entier, correspondant au numéro du jour ;

**nomMois** est de type chaîne de 9 caractères, correspondant au nom du mois ;

**numeroAnnée** est de type entier, correspondant au numéro de l'année

*FinEnregistrement*

# Enregistrements

**ADRESSESOC** = *Type Enregistrement regroupant*

**nomSoc** est de type chaîne de caractères, correspondant au nom de la société ;

**numEntree** est de type entier, correspondant au numéro de l'entrée sur la rue ;

**nomRue** est de type chaîne de caractères, correspondant au nom de la rue ;

**bp** est de type entier, correspondant au numéro de la boîte postale ;

**cp** est de type entier, correspondant au code postal ;

**nomVille** est de type chaîne de caractères, correspondant au nom de la ville ;

**nomPays** est de type chaîne de caractères, correspondant au nom du pays

*FinEnregistrement*

# Enregistrements

Déclaration des variables de type enregistrement

Nom\_variable est de type NomTypeEnregistrement,  
correspondant à ...

Exemples :

**depart** est de type DATE, correspondant à la date de départ en congés ;

**adressClient** est de type ADRESSESOC, correspondant à l'adresse  
de la société en cours de traitement

# Enregistrements

Référence aux champs des variables de type enregistrement

NomVariable•NomChamp

Exemples :

**depart.nomJour** pour désigner le nom du jour de la variable  
**depart**

**adressClient.bp** pour désigner la boite postale de la variable  
**adressClient**

# Enregistrements

Possibilités de déclaration de tableaux comme champs dans un type enregistrement

Possibilités d'utilisation de tableaux d'enregistrements

Possibilités d'utilisation de structures complexes (liste, files, fichiers, etc.) dont les éléments sont de type enregistrement

# Enregistrements

Exercice : On veut gérer les étudiants inscrits au cours d'algorithmique et structures de données statiques.

- 1) Lister les informations associées à un étudiant.
- 2) Décrire la structure de données qu'il faudrait employer.
- 3) Ecrire un algorithme qui initialise les données concernant un étudiant.
- 4) Ecrire un algorithme qui initialise les données des N étudiants inscrits au cours d'algorithmique et structures de données statiques.

# Enregistrements

- 1) Nom ← Chaîne de 30 caractères
- 2) Prénom ← Chaîne de 50 caractères
- 3) Genre ← Caractère
- 4) Année de naissance ← Entier naturel
- 5) Lieu de naissance ← Chaîne de 30 caractères
- 6) Numéro de téléphone ← Chaîne de 20 caractères
- 7) Email ← Chaîne de 50 caractères
- 8) Filière ← Chaîne de 4 caractères
- 9) Niveau d'étude ← Entier naturel
- 10) Numéro secteur d'habitation ← Entier naturel
- 11) Ville d'habitation ← Chaîne de 20 caractères
- 12) Boîte postale ← Chaîne de 10 caractères
- 13) Statut travail ← caractère
- 14) Nationalité ← Chaîne de 15 caractères
- 15) Pays de résidence ← Chaîne de 30 caractères

# Enregistrements

**ADRESSE** = *Type Enregistrement regroupant*

NumTel est de type chaîne de 20 caractères, correspondant à ...

Email est de type chaîne de 50 caractères, correspondant à ...

NumSecteur est de type entier naturel, correspondant à ...

Ville est de type chaîne de 20 caractères, correspondant à ...

Bp est de type chaîne de 10 caractères, correspondant à ...

Pays est de type chaîne de 30 caractères, correspondant à ...

*FinEnregistrement*

# Enregistrements

**ETUDIANT = Type Enregistrement regroupant**

Nom est de type chaîne de 30 caractères, correspondant à ...

Prenom est de type chaîne de 50 caractères, correspondant à ...

Genre est de type caractère, correspondant à ...

AnNais est de type entier, correspondant à ...

LieuNais est de type chaîne de 30 caractères, correspondant à ...

Filiere est de type chaîne de 4 caractères, correspondant à ...

Niveau est de type entier naturel, correspondant à ...

StatuTravail est de type caractère, correspondant à ...

Nationalite est de type chaîne de 15 caractères, correspondant à ...

Adr est de type ADRESSE, correspondant à ...

*FinEnregistrement*

# Enregistrements

## Exemples de déclarations de variables

toto est de type ADRESSE, correspondant à ...

bonEtudiant est de type ETUDIANT, correspondant à ...

dernier est de type ETUDIANT, correspondant à ...

## Exemples d'initialisation des champs des variables

```
toto.numTel← "70726931" ;
```

```
toto.Email← "tmesmin.dandjinou@univ-bobo.bf" ;
```

```
toto.numSecteur← 28
```

# Enregistrements

## Exemples de déclarations de variables

toto est de type ADRESSE, correspondant à ...

bonEtudiant est de type ETUDIANT, correspondant à ...

dernier est de type ETUDIANT, correspondant à ...

## Exemples d'initialisation des champs des variables

bonEtudiant.Nom← "OUEDRAOGO" ;

bonEtudiant.Prenom← "WENDPANGA" ;

bonEtudiant.Genre← 'H' ;

bonEtudiant.Adr.NumTel← "20202020" ;

bonEtudiant.Adr.Ville← "Bobo-Dioulasso"

# Enregistrements

## Exemples d'initialisation des champs des variables

```
dernier.Nom← "ZIRCONGO" ;  
dernier.Prenom← "Sannom" ;  
dernier.Genre← 'H' ;  
dernier.Adr.NumTel← "50505050" ;  
dernier.Adr.Ville← "Ouagadougou"
```

## T : tableau [1..6] ETUDIANT

```
T(1).Nom← "SARE" ;  
T(1).Prenom← "Oussou" ;  
T(2).Nom← "ZAN" ;  
T(2).Prenom← "Moussa"
```

**FIN**

**QUESTIONS ?**