

Chapitre 3

Les Tableaux



- 1) Introduction
- 2) Définition
- 3) Tableau à une dimension
- 4) Tableau à deux dimensions



1. Introduction

Supposons que l'on veut saisir quatre notes de quatre étudiants et de les afficher après avoir multiplié toutes les notes par sept.

Algorithme **Note**

Variables N1, N2, N3, N4 : Réel; /* Déclaration des variables */

Début

Écrire(" Entrez la valeur de la première note ");

Lire(N1);

Écrire(" Entrez la valeur de la deuxième note ");

Lire(N2);

Écrire(" Entrez la valeur de la troisième note ");

Lire(N3);

Écrire(" Entrez la valeur de la quatrième note ");

Lire(N4);

Écrire(" La note 1 multipliée par 7 est :"; N1 *7);

Écrire(" La note 2 multipliée par 7 est :"; N2 *7);

Écrire(" La note 3 multipliée par 7 est :"; N3 *7);

Écrire(" La note 4 multipliée par 7 est :"; N4 *7);

Fin



Problème

Supposons qu'on veut stocker les notes de 60 étudiants, alors nous avons besoin de déclarer 60 variables (N1,N2,.....,N60). De plus, la même instruction va se répéter 60 fois. Cette façon n'est pas très pratique.

Solution

Les langages de programmation offrent la possibilité de rassembler toutes ces variables dans une seule structure de donnée appelée **tableau** qui est facile à manipuler.



2. Définition

- ❑ **Un tableau** est une suite d'éléments de même type. Il utilise plusieurs cases mémoire à l'aide d'un **seul nom**.
- ❑ Comme toutes les cases portent le même nom, une variable entière nommée **indice** permet d'indiquer la **position** d'un élément donné au sein du tableau et de déterminer sa **valeur**.
- ❑ Nous allons étudier deux types de tableaux :
 - Tableau à une dimension (ou vecteur)
 - Tableau à deux dimensions (ou matrice)



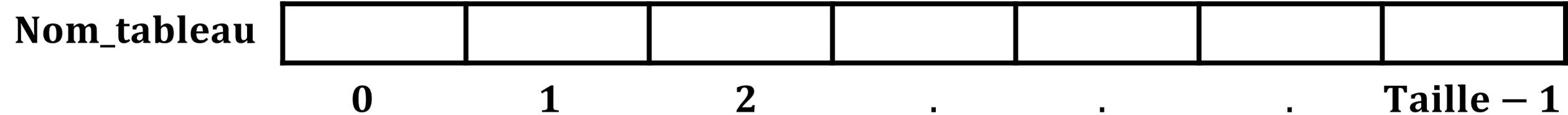
3. Tableaux à une dimension

3.1 Déclaration

La déclaration d'un tableau permet d'associer à un nom une zone mémoire composée d'un certain nombre de cases de même type.

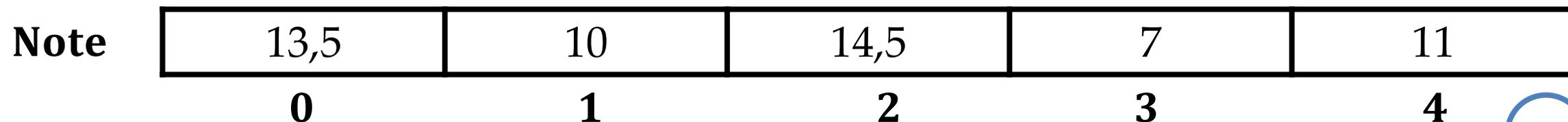
Syntaxe : Variable **tableau** Nom_tableau [taille]: type

La représentation schématique d'un tableau dans la mémoire est comme suit :



Exemple : Variable **tableau** Note[5] : réel;

La représentation schématique d'un tableau dans la mémoire est comme suit :





3. Tableaux à une dimension

- ❑ Ce **tableau** est de longueur **5**, car il contient **5 emplacements**.
- ❑ Chacun des cinq **valeurs** du tableau est **repéré** par son rang, appelé **indice**.
- ❑ Pour **accéder** à un élément du tableau, il suffit de préciser entre crochets l'**indice** de la **case** contenant cet élément. Pour accéder au **3ème** élément (**14,5**), on écrit : **Note[2]**. Pour accéder au $i^{\text{ème}}$ élément, on écrit **Note[i-1]** (avec $0 \leq i < 5$).

Remarques:

- ❑ Le premier élément d'un tableau porte l'indice **0** ou l'indice **1** selon les langages. Le plus souvent c'est 0 (c'est ce qu'on va utiliser).
- ❑ On peut définir des tableaux de tous types : entiers, réels, caractères, booléens, chaînes de caractères, ...
- ❑ La valeur d'un indice doit être inférieur ou égale au nombre d'éléments du tableau.



3. Tableaux à une dimension

3.2 Accès aux éléments d'un tableau

Les instructions d'affectation, de lecture, et d'écriture s'appliquent aux tableaux comme aux variables.

Syntaxe:

- ❑ **Affectation** : `Nom_tableau[indice]←Valeur;`
- ❑ **Lecture** : `Lire (Nom_tableau[indice]);`
- ❑ **L'écriture** : `Ecrire (Nom_tableau[indice]);`

Exemples:

- ❑ `Note[3]←15`: Cette instruction permet d'affecter la valeur **15** au **4ème** élément du tableau.
- ❑ `Lire(Note[2])`: Cette instruction permet de saisir une valeur introduite par l'utilisateur à l'élément **trois** du tableau Note.



3. Tableaux à une dimension

❑ **Ecrire(Note[3]):** Cette instruction affiche le contenu du **quatrième** élément du tableau Note.

Remarque :

Il ne faut pas confondre l'**indice** d'un élément d'un tableau avec son **contenu**

Exercice d'application 1:

- 1) Déclarer un tableau T de 5 éléments de type réel ?
- 2) Donner la représentation schématique du tableau T après l'exécution des instructions suivantes :

$T[0] \leftarrow 10;$

$T[1] \leftarrow T[0]+4;$

$T[2] \leftarrow T[1]*2-11;$

$T[3] \leftarrow T[1]+T[2];$

$T[4] \leftarrow T[1] \bmod 10 ;$



3. Tableaux à une dimension

Correction:

La représentation schématique d'un tableau dans la mémoire est comme suit :

| | | | | | |
|----------|----------|----------|----------|----------|----------|
| T | 10 | 14 | 17 | 31 | 4 |
| | 0 | 1 | 2 | 3 | 4 |

Exercice d'application 2:

Ecrire un algorithme qui permet de remplir un tableau avec les notes de 7 étudiants, ensuite modifier la note de l'étudiant numéro 3 par une autre note (12,50 par exemple), puis afficher les notes des 4 derniers étudiants ?

Correction:

Algorithme Tableau_note
Variable **tableau** Note[7]: réels;

Début



3. Tableaux à une dimension

```
Lire(Note[0]);  
Lire(Note[1]);  
Lire(Note[2]);  
Lire(Note[3]);  
Lire(Note[4]);  
Lire(Note[5]);  
Lire(Note[6]);
```

/*Ces instructions permet de saisir
les notes des 7 étudiants */

```
Note[2] ← 12,5; /*Modifier la troisième note du tableau */
```

```
Ecrire(Note[3]);  
Ecrire(Note[4]);  
Ecrire(Note[5]);  
Ecrire(Note[6]);
```

/*Ces instructions permet d'afficher
les notes des 4 derniers étudiants */

Fin



3. Tableaux à une dimension

3.3 Parcours d'un tableau

Pour **parcourir** les **éléments** du **tableau** selon l'ordre **croissant** (ou **décroissant**) des **indices**, on utilise des **boucles**, et seule la valeur de l'**indice** est amenée à **changer**.

Exemple 1:

Écrire un algorithme permettant de saisir 30 notes et de les afficher après avoir multiplier ces notes par un coefficient fourni par l'utilisateur ?

Solution :

```
Algorithme    Tableau_note
Variable tableau Note[30]: réels;
Variables Coef,i : entiers;
Début
  Écrire("Entrez le coefficient");
  Lire(Coef);
  Pour i ← 1 jusqu'à 30 Faire
    Écrire("Entrez la valeur de la note ", i);
    Lire(Note[i-1]);
    Écrire(Note[i-1]*Coef);
  FinPour
```

Fin



3. Tableaux à une dimension

Exemple 2:

Écrire un algorithme qui calcule la somme des éléments d'un tableau ? On suppose que le nombre d'éléments du tableau ne dépasse pas 100.

Solution :

Algorithme Somme_Tableau

Variable **tableau** tab[100]: réels;

Variables **N,i** : entiers;

Variable **S** : réel;

Début

$S \leftarrow 0;$

Pour $i \leftarrow 1$ jusqu'à 100 **Faire**

Écrire("Entrez la valeur de l'élément ", i ,"du tableau);

Lire($T[i-1]$);

$S \leftarrow S + T [i-1];$

FinPour

Écrire("La somme des éléments de tableau est : " , S);

Fin



3. Tableaux à une dimension

Exemple 3:

Que produit l'algorithme suivant ?

Algorithme Suite_de_Fibonacci
Variable **tableau** F[10],i: entiers;

Début

F[0] ← 0;

F[1] ← 1;

Écrire(F[0],F[1]);

Pour i ← 2 jusqu'à 9 Faire

F[i] ← F[i-1]+F[i-2];

Ecrire(F[i]);

FinPour

Fin

Solution:

Contenu

Indice

F

| | |
|----|---|
| 0 | 0 |
| 1 | 1 |
| 1 | 2 |
| 2 | 3 |
| 3 | 4 |
| 5 | 5 |
| 8 | 6 |
| 13 | 7 |
| 21 | 8 |
| 34 | 9 |



4. Tableaux à deux dimension

Reprenons l'exemple précédent et supposons qu'un étudiant a plusieurs notes (une note par matière). Pour quatre étudiants nous aurons le tableau de relevé de notes suivant :

| | Ahmed | Rania | Nada | Karim |
|-------------|-------|-------|-------|-------|
| Electricité | 12,00 | 10,50 | 07,50 | 11,00 |
| Algorithmme | 15,50 | 17,00 | 13,50 | 08,00 |
| Mécanique | 16,50 | 09,00 | 13,00 | 10,50 |

Les **tableau à deux dimensions** se représentent comme **une matrice** ayant un certain nombre de **lignes** (première dimension), et un certain nombre de **colonnes** (deuxième dimension).



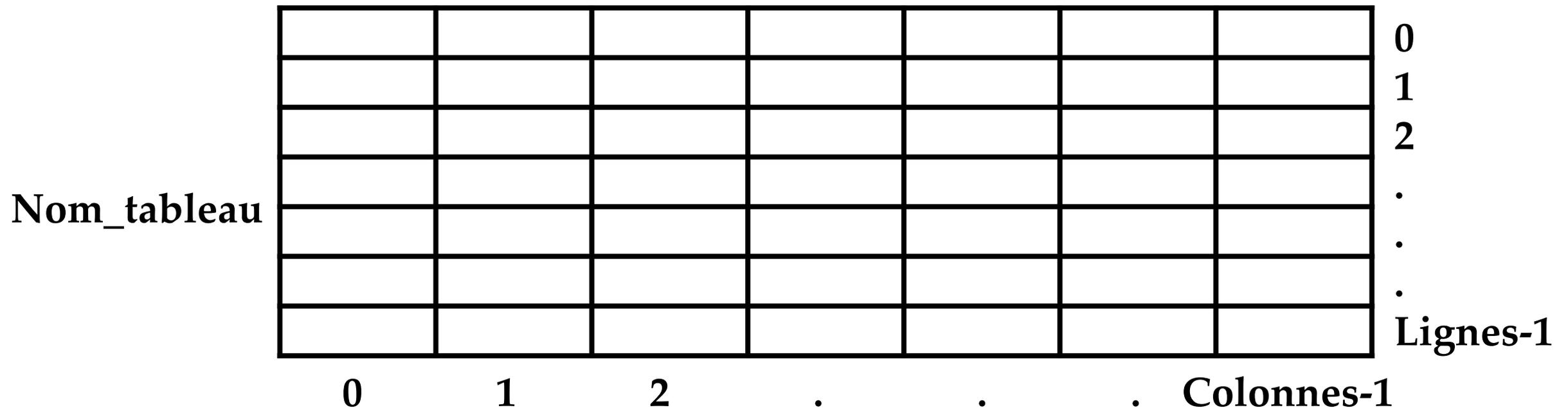
4. Tableaux à deux dimension

4.1 Déclaration

Syntaxe :

Variable **tableau** Nom_tableau [**lignes**][**colonnes**]: type

La représentation schématique d'un tableau à deux dimensions dans la mémoire est comme suit :





4. Tableaux à deux dimension

Exemple: Déclarer un tableau de notes de type réel à deux dimensions composé de 3 lignes et de 4 colonnes?

Réponse : Variable **tableau** Notes [3][4]: réels;

| | | | | | |
|-------|-------|-------|-------|-------|---|
| | 13,50 | 15,00 | 16,50 | 17,00 | 0 |
| Notes | 16,00 | 04,50 | 07,50 | 13,50 | 1 |
| | 13,00 | 17,50 | 12,00 | 18,50 | 2 |
| | 0 | 1 | 2 | 3 | |

Ce tableau a 3 lignes et 4 colonnes. Les éléments du tableau sont repérés par leur numéro de ligne et leur numéro de colonne. Par exemple Notes[1][0] vaut 16,00.



4. Tableaux à deux dimension

4.2 Accéder à un élément

Pour accéder à un élément de la matrice (tableau à deux dimensions), il suffit de préciser entre crochets les **indices** de la case contenant cet élément.

Syntaxe: **Affectation** : `Nom_tableau[num_ligne][num_colonne]←Valeur;`

Lecture : `Lire (Nom_tableau[num_ligne][num_colonne]);`

L'écriture : `Ecrire (Nom_tableau[num_ligne][num_colonne]);`

Exemples:

`Notes[2][3]←13`: Cette instruction permet d'affecter la valeur **13** pour la **troisième** matière (Mécanique) à l'étudiant numéro **quatre** (Karim) dans le tableau **Notes**.

`Lire(Notes[1][0])`: Cette instruction permet de saisir une note de la **deuxième** matière (Algorithme) pour l'étudiant numéro **un** (Ahmed) du tableau **Notes**.

`Ecrire(Notes[0][2])`: Cette instruction affiche la note de la **première** matière (Electricité) pour l'étudiant numéro **trois** (Nada) dans le tableau **Notes**.



4. Tableaux à deux dimension

4.3 Parcours complet d'un tableau à deux dimensions (matrice)

Pour parcourir une matrice, nous avons besoin de deux **boucles**, l'une au sein de l'autre, c'est ce qu'on appelle des **boucles imbriquées**. La **première** par exemple est pour parcourir **les lignes** et la **deuxième** est conçue pour parcourir les **colonnes**.

Exemple :

Écrire un algorithme qui permet de saisir et d'afficher les notes d'une classe de 30 étudiants en 5 matières ?

Solution:

Algorithme Note_matière
Constante NL=5;
Constante NC=30;
Variable **tableau** Note[NL][NC]: réels;
Variables **i,j** : entiers;



4. Tableaux à deux dimension

Début

Pour $i \leftarrow 1$ jusqu'à NL Faire

Pour $j \leftarrow 1$ jusqu'à NC Faire

Écrire("Entrez la note de l'étudiant " , j , "dans la matière " , i);

Lire(Note[$i-1$][$j-1$]);

Ecrire(Note[$i-1$][$j-1$]);

FinPour

Ecrire("\n");

FinPour

Fin



4. Tableaux à deux dimension

Exercice d'application:

Écrire un algorithme qui permet de demander à l'utilisateur de saisir les notes des étudiants (5 étudiants) dans chaque matière (3 matières), puis l'algorithme calcul et affiche la moyenne de chaque étudiant.

Solution:

Algorithme Note_matière

Constante NL=5;

Constante NC=3;

Variable **tableau** Note[NL][NC]: réels;

Variables **i, j, S** : entiers;



4. Tableaux à deux dimension

Début

Pour $i \leftarrow 0$ jusqu'à $NL-1$ Faire

$S \leftarrow 0$;

Pour $j \leftarrow 0$ jusqu'à $NC-1$ Faire

Lire (Note[i][j]);

$S \leftarrow S + \text{Note}[i][j]$;

FinPour

$\text{Moy} \leftarrow S/3$;

Écrire ("La moyenne de l'étudiant", $i+1$, "est", Moy);

FinPour

Fin



TD4: Les tableaux

Exercice 1:

En utilisant les tableaux, écrire un algorithme qui permet la saisie d'une liste de 60 moyennes réelles et d'afficher le nombre de moyennes supérieures ou égales à 10.

Exercice 2 (Permutation des éléments d'un tableau):

Écrire un algorithme qui permute les éléments d'un tableau de 8 éléments en plaçant le dernier en premier et ainsi de suite, voir exemple suivant.

| | | | | | | | |
|-----------------|----|----|----|----|----|----|----|
| 12 | 34 | 5 | 43 | 55 | 56 | 23 | 1 |
| Tableau initial | | | | | | | |
| 1 | 23 | 56 | 55 | 43 | 5 | 34 | 12 |
| Tableau final | | | | | | | |



Exercice 3 (Maximum des moyennes d'une classe):

Écrire un algorithme qui permet la saisie des moyennes d'une classe de 10 étudiants et affiche le maximum ?

Exercice 4 (Trier un tableau):

Écrire un algorithme qui permet la saisie de 4 moyennes et de les classer selon l'ordre croissant ?

Exercice 5 (Tableau à deux dimensions)

Écrire un algorithme qui permet de :

- Saisir les notes d'une classe de 15 étudiants en 4 matières ;
- Calculer et afficher la moyenne de chaque étudiant ;
- Calculer et afficher la moyenne de la classe dans chaque matière ;
- Calculer et afficher la moyenne générale de la classe.



Exercice 6 (La plus grande valeur):

Soit un tableau T à deux dimensions de 12 lignes et 8 colonnes préalablement rempli de valeurs numériques. Écrire un algorithme qui recherche la plus grande valeur au sein de ce tableau.

Exercice 7 (Somme de deux matrices):

Écrire un algorithme qui calcul la somme de deux matrices A et B préalablement rempli de valeurs numériques ?

Exercice 8 (Produit de deux matrices):

Écrire un algorithme qui calcul le produit de deux matrices A(2,3) et B(3,2) remplies au préalable ?