

# Chapitre 4 :

# Les Tableaux

1

## Exemple Introductif

- Imaginons que l'on veuille calculer **la moyenne des notes** d'une promotion, **quel algorithme** allons nous utiliser ?
- Pour l'instant on pourrait avoir l'algorithme suivant :

### Algorithme MoyennePromotion

**Variables** somme, note: Réel  
nbEleves, i: Entier

#### Début

Lire (nbEleves)

somme  $\leftarrow$  0

Pour i = 1 A nbEleves

Ecrire("Saisir la note de l'élève numéro", i)

lire(note)

somme  $\leftarrow$  somme+note

FinPour

Ecrire ("La moyenne des notes est :",somme/ nbEleves)

#### Fin

## Exemple Introductif

- Imaginons que l'on veuille toujours calculer **la moyenne des notes** d'une promotion mais **en gardant en mémoire toutes les notes des étudiants** (pour par exemple faire d'autres calculs tels que l'écart type, la note minimale, la note maximale, etc.)
- **Il faudrait alors déclarer autant de variables** qu'il y a d'étudiants. Par exemple en supposant qu'il y ait 3 étudiants

```
Variables somme, note1, note2, note3: Réel  
           i: Entier  
Début  
    Ecrire ("Saisir les notes des trois étudiants :")  
    Lire (note1)  
    Lire (note2)  
    Lire (note3)  
    somme ← note1+note2+note3  
    Ecrire ("La moyenne est de :",somme/3)  
Fin
```

## Exemple Introductif

- **Le problème** est lorsque le nombre d'étudiants est élevé
  - Si on en a 10, il faut déclarer 10 variables
  - Si on en a  $n$ , il faut déclarer  $n$  variables
  - . . . **Ce n'est pas réaliste**
- Heureusement, **les langages de programmation** offrent la possibilité de **rassembler toutes ces variables** dans une seule structure de donnée appelée **tableau**

# Tableaux unidimensionnels

- Un **tableau** est un **ensemble d'éléments de même type désignés par un identificateur unique**
- La **déclaration d'un tableau** s'effectue en précisant **le type** de ses éléments et sa **dimension/taille** (le nombre de ses éléments)
- **Syntaxe:**

**Tableau** identificateur **[dimension]** : **type**

- **identificateur** : désigne le nom du tableau
  - **dimension** : désigne le nombre d'éléments du tableau. On dit aussi sa taille
  - **type** : c'est le type du tableau (le type de tous ces éléments)
- On peut définir des tableaux de tous types : tableaux d'entiers, de réels, de caractères, de booléens, de chaînes de caractères, ...

# Tableaux unidimensionnels

- Exemples : Tableaux à une dimension (unidimensionnels)

- Tableau Note [6] : Réel

|      |    |   |   |      |    |       |
|------|----|---|---|------|----|-------|
| Note | 12 | 5 | 8 | 13,5 | 20 | 16,25 |
|      | 0  | 1 | 2 | 3    | 4  | 5     |

Contenu de la case numéro 3

numéro de la case

- Note est un tableau qui contient six valeurs réelles (6 éléments)
- Chacun des 6 nombres du tableau est repéré par son rang, appelé indice.
- Tableau nom [10] , prenom [10] : Chaîne**
  - nom et prenom sont deux tableaux de 10 éléments de type chaîne

# Tableaux unidimensionnels

- On peut accéder (**en lecture ou en écriture**) à un élément d'un tableau à partir de son indice (ou rang) en utilisant **la syntaxe** suivante :

**nom\_tableau [indice]**

- les cases du tableau sont numérotées successivement. Ce numéro est appelé indice. Le premier indice du tableau est soit 0, soit 1. La case suivante porte le numéro suivant, etc. Le plus souvent c'est 0 (c'est ce qu'on va utiliser en pseudo-code).
- Dans ce cas:
  - `notes[i]` : désigne l'élément  $i+1$  du tableau `notes`. Par exemple **`notes[3]`** désigne le 4<sup>ème</sup> élément du tableau **`notes`**
  - L'indice doit être compris entre 0 et Taille-1 (bornes incluses)

# Tableaux unidimensionnels

- Dans un **tableau**, la valeur d'un **indice** doit toujours :
  - être un nombre entier. Quel que soit le langage, l'élément `nom_tableau[3,1416]` n'existe jamais.
  - être inférieure ou égale au nombre d'éléments du tableau (moins 1, si l'on commence la numérotation à zéro)
- Il faut différencier le numéro de la case (**indice**) de ce qu'il y a dans la case (`nom_tableau [indice]`).



# Tableaux unidimensionnels

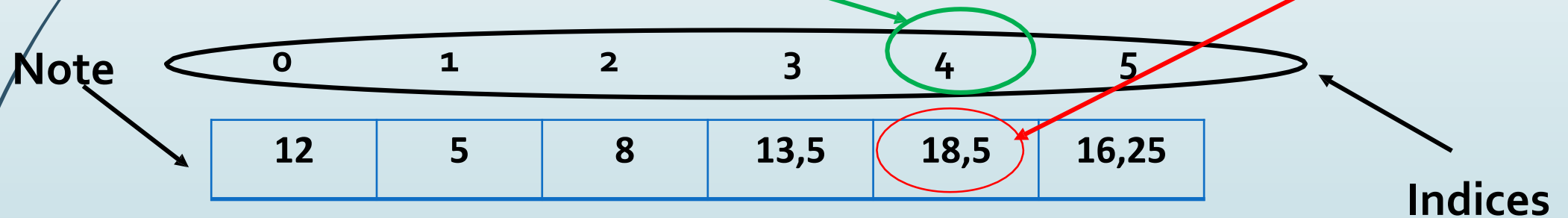
## Exemple d'accès à un élément :

- Si X est un tableau de 10 entiers :
  - **$X[2] \leftarrow -5$** : met la valeur -5 dans la **case numéro 2** du tableau
  - **Lire  $X[1]$** : met l'entier saisi par l'utilisateur dans la case numéro 1 du tableau X
  - **Ecrire  $X[1]$** : affiche la valeur de la case numéro 1 du tableau X
- En considérant le cas où a est une variable de type Entier,
  - **$a \leftarrow X[2]$** : met la valeur de la case numéro 2 du tableau X dans a

# Tableaux unidimensionnels

## Exemple

- Le tableau a :
  - un nom : Note (dans notre exemple),
  - un nombre d'éléments 6 qui sont chacun désignés par un indice (ex : 18,5 sera désigné par l'indice 4)



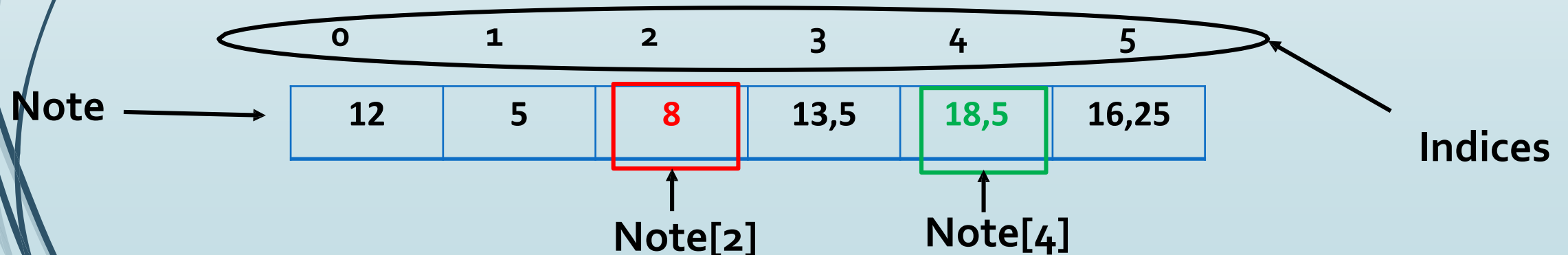
# Tableaux unidimensionnels

- **L'indice** peut être :
  - une constante : Note [2] vaut 8
  - une variable :  $\text{ind} \leftarrow 3$  ; Note [ind] vaut 13,5
  - une expression : Note [ind + 1] vaut 18,5
- **Règle générale** : un élément de tableau se comporte comme une variable simple et peut faire partie d'une expression. On peut trouver un élément de tableau :
  - dans une instruction de lecture : lire(Note[4])
  - dans une instruction d'écriture : Ecrire(Note[4])
  - à gauche d'une flèche d'affectation : Note[5]  $\leftarrow$  45
  - dans une expression : Ecrire(Note[5] \* 10)
  - Etc.

# Tableaux unidimensionnels

## ▪ Exemple

- Pour accéder aux éléments du tableau, on va utiliser le nom du tableau et les indices : on notera `Note [2]` l'élément contenu dans le tableau **Note** à l'indice 2
- `Note [2]` est considérée comme une variable simple :
  - Lire (`Note[2]`) permet de mettre une valeur dans la case du tableau `Note` à l'indice 2
  - l'expression `Note[2] + Note[4]` aura comme valeur 26,5



# Tableaux unidimensionnels

- Exercice: Donner les représentations graphiques des tableaux X et voyelle après l'exécution des programmes suivants

Tableau X [4] : Entier

DEBUT

X [0] ← 12

X [1] ← 5

X [2] ← 8

X [3] ← 20

FIN

Tableau voyelle [6] : Chaîne

DEBUT

voyelle [0] ← "a"

voyelle [1] ← "e"

voyelle [2] ← "i"

voyelle [3] ← "o"

voyelle [4] ← "u"

voyelle [5] ← "y"

FIN

# Tableaux unidimensionnels

**Manipulation d'un tableau :** Les instructions de lecture, écriture et affectation s'appliquent aux éléments de tableaux comme aux variables. Alors, Une fois déclaré, un tableau peut être utilisé comme un ensemble de variables simples.

- **L'affectation :**

- Pour affecter une valeur à un élément  $i$  d'un tableau nommé par exemple  $A$ , on écrira :  $A[i] \leftarrow \text{valeur}$ .
- **Par exemple :** l'instruction :  $A[0] \leftarrow 20$  ; affecte au premier élément du tableau  $A$  la valeur 20.
- Pour affecter la même valeur à tous les éléments d'un tableau  $A$  de type numérique et de dimension 10, on utilise une boucle :

```
Pour i = 0 à 9  
    A[i] ← 0  
FinPour
```

# Tableaux unidimensionnels

- **La lecture :**

- Comme les variables simples, il est possible aussi d'assigner des valeurs aux éléments d'un tableau lors de l'exécution c.-à-d. les valeurs sont saisies par l'utilisateur à la demande du programme.

- Exemple :

```
Ecrire ("Enter une note :")  
Lire A[6]
```

- Dans cet exemple, la valeur saisie est affectée au 7ème élément du tableau A.

- **L'écriture:**

- De façon analogue à la lecture, l'écriture de la valeur d'un élément donné d'un tableau s'écrit comme suit : Ecrire (A [i] )
- Cette instruction permet d'afficher la valeur de l'élément  $i+1$  du tableau A.

# Tableaux unidimensionnels

- **Un grand avantage** des **tableaux** est qu'on peut **traiter les données** qui y sont stockées de façon simple en utilisant **des boucles**
- Pour parcourir les éléments du tableau selon **l'ordre croissant** (ou **décroissant**) des **indices**, on utilise des **boucles**
- **Exemple** : Au lieu d'écrire 5 instructions de lecture pour saisir les valeurs du tableau :

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

- Une boucle à 5 itérations ferait aussi bien l'affaire :

```
Pour i = 0 à 4  
    Lire (Note [i])  
FinPour
```



# Tableaux unidimensionnels

Saisie et affichage des éléments d'un tableau :

- Saisie d'un tableau

```
Pour i = 0 à N-1  
    Ecrire ("Saisie de l'élément ", i + 1)  
    Lire (T[i])  
FinPour
```

- Affichage d'un tableau

```
Ecrire ("Le tableau saisi est")  
Pour i = 0 à N-1  
    Ecrire (T[i], " ")  
FinPour
```

# Tableaux unidimensionnels

**Initialisation des éléments d'un tableau :**

Le bloc d'instructions suivant initialise un à un tous les éléments d'un tableau de N éléments :

```
Pour i = 0 à N-1  
    T[i] ← 0  
FinPour
```

## Exemple

### Algorithme Exemple

Variable i : entier

Tableau T[12] vecteur d'entiers

#### Début

T[10] <- 0

T[11] <- 1

Pour i  $\equiv$  1 à 10

Lire T[i]

T[10] <- T[10] + T[i]

T[11] <- T[11] \* T[i]

Finpour

Ecrire T[10]

Ecrire T[11]

#### Fin

## Exercices

- Q1: Quel résultat fournira l'exécution de ce programme : Si on saisit successivement les valeurs : 2, 5, 3, 10, 4, 2.

```
Variables i: Entier
Tableau C[6]: Entier
DEBUT
    POUR i = 0 A 5
        Lire (C[i] )
    FINPOUR
    POUR i = 0 A 5
        C[i] ← C[i] * C[i]
    FINPOUR
    POUR i = 0 A 5
        Ecrire (C[i], "\n")
    FINPOUR
FIN
```

4

25

9

100

16

4

## Exercices

- Q2: Quel résultat fournira l'exécution de ce programme

Variable i : Entier

Tableau suite [8] : Entier

DEBUT

Suite [0]  $\leftarrow$  1

Suite [1]  $\leftarrow$  1

POUR i = 2 A 7

    suite [i]  $\leftarrow$  suite [i - 1] + suite [i - 2]

FINPOUR

POUR i = 0 A 7

    Ecrire (suite [i], "\n")

FINPOUR

FIN

1  
1  
2  
3  
5  
8  
13  
21

## Exercices

- Q3: Quel résultat fournira l'exécution de ce programme ? Peut-on simplifier cet algorithme avec le même résultat ?

```
Variable i : Entier
Tableau Nb[6] :Entier
Début
    Pour i =0 à 5
        Nb(i) ← i * i
    FinPour
    Pour i = 0 à 5
        Ecrire Nb(i)
    FinPour
Fin
```

- **Q4:** Ecrire un algorithme qui demande à l'utilisateur de saisir 100 notes au clavier. Ensuite, il les affiche et calcule leur moyenne
- **Q5:** Ecrire un algorithme pour le calcul du nombre d'étudiants ayant une note supérieure strictement à 10

- Q6 : Ecrire un algorithme qui demande à l'utilisateur de saisir 20 valeurs dans un tableau et il affiche l'inverse du tableau
- Q7 : Ecrire un algorithme qui demande à l'utilisateur de remplir la quantité et le prix et calcule le total

|          |      |    |      |    |    |    |      |    |    |    |     |
|----------|------|----|------|----|----|----|------|----|----|----|-----|
| Quantité | 5    | 8  | 10   | 4  | 1  | 2  | 1    | 3  | 4  | 8  |     |
| Prix     | 1,50 | 3  | 4,50 | 20 | 10 | 47 | 9,50 | 6  | 8  | 10 |     |
| Total    | 7,50 | 24 | 45   | 80 | 10 | 94 | 9,50 | 18 | 32 | 80 | 400 |