

# **Chapitre 3**

### Algorithmique & Programmation en C

- Tableau 1D
- Tableau 2D

## **Tableaux**



- Un ensemble d'éléments de <u>même type</u> désigné par un <u>identificateur unique</u>.
- Chaque élément est repéré par un <u>indice</u> précisant sa position au sien de l'ensemble.
- Caractéristiques d'un tableau:
  - Identificateur
  - Type
  - Taille
  - Indice

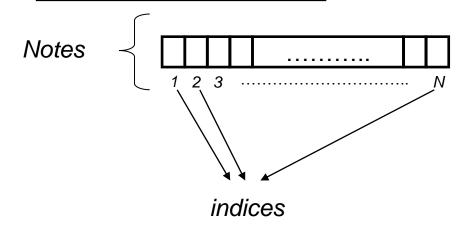
## **Tableaux**



# Tableau 1D



### • Tableau à un indice



- Identificateur du tableau: *Notes*
- Dimension du tableau *Notes*: *N*
- Indice du premier élément: 1
- Indice du dernier élément: N
- Contenu du 1 élément: *Notes(1)*
- Contenu du 2 élément: Notes (2)

- ..



#### **Déclaration:**

Tableau Nom\_tableau(taille):Type;

### **Exemple:**

Tableau Notes(20):Entier;

Tableau tab(10):Réel;



#### Lecture:

On considère les déclarations suivantes:

- tab: Nom du tableau

- taille: la taille du tableau

- id: l'indice

```
Ecrire('Saisir le tableau: ');
Pour id←1 à taille pas 1
Lire(tab(id));
finpour
```



#### Affichage:

On considère les déclarations suivantes:

- tab: Nom du tableau

- taille: la taille du tableau

- id: l'indice



## **Exercice 1**

Ecrire un algorithme qui permet de saisir et d'afficher les éléments d'un tableau.



# Exercice 1 (Solution)

```
Algorithme Tableau1D;
 Tableau T(50): Entier;
 Variable N, i: Entier;
Début
 Ecrire ('Entrer la taille:');
 Lire (N);
 Ecrire ('Saisir le tableau:');
 pour i←1 à N pas 1
    Lire (T(i));
 finpour
```

```
Ecrire ('Afficher le tableau:');
 pour i←1 à N pas 1
     Ecrire (T(i));
 finpour
Fin
```



#### **Recherche:**

On considère les déclarations suivantes:

- tab: Nom du tableau
- taille: la taille du tableau
- id: l'indice
- e : l'élément à rechercher
- pos : la position de e dans le tableau

```
pos ← -1;
Pour id←1 à taille pas 1
si tab(id)=e alors
pos ← id;
finsi
finpour
```



## **Exercice 2**

Ecrire un algorithme qui permet de:

- Lire la taille du tableau (N)
- Lire les éléments du tableau (A)
- Afficher le contenu du tableau (A)
- Lire l'élément à rechercher (elt)
- Rechercher (elt) dans (A) en affichant les indices



# Exercice 2 (Solution)

```
Algorithme Tableau1D;
 Tableau A(50): Entier;
 Variable N, elt, i: Entier;
Début
 Ecrire ('Entrer la taille:');
 Lire (N);
 Ecrire ('Saisir le tableau:');
 pour i←1 à N pas 1
    Lire (T(i));
 finpour
```

```
Ecrire ('Afficher le tableau:');
pour i←1 à N pas 1
    Ecrire (T(i));
finpour
Ecrire ('Entrer elt à rechercher:');
Lire (elt);
pour i←1 à N pas 1
 si A(i) = elt alors
   Ecrire ('Indice:',i);
 finsi
Finpour
Fin
                                       12
```

Prof: R. EL AYACHI



#### **Suppression:**

On considère les déclarations suivantes:

- tab: Nom du tableau

- taille: la taille du tableau

- id: l'indice

- e : l'élément à supprimer du tableau

- pos : la position de e dans le tableau

tab:

6 7 8 9 10

e:

8

1- Rechercher

2- Supprimer



### **Recherche:**

### **Suppression:**

```
pos ← -1;
pour id←1 à taille pas 1
si tab(id)=e alors
pos ← id;
finsi
finpour
```

```
si pos <> -1 alors

Pour id←pos à taille-1 pas 1

tab(id) ← tab(id+1);

finpour

taille ← taille -1;

finsi
```

```
e: 8 tab: 6 7 8 9 10
```

```
tab: 6 7 9 10 10
```

14



## **Exercice 3**

Ecrire un algorithme qui permet de:

- Lire la taille du tableau (N)
- Lire les éléments du tableau (A)
- Afficher le contenu du tableau (A)
- Lire l'élément à supprimer (elt)
- Supprimer (elt) dans le tableau (A)
- Afficher le tableau après la suppression (A)<sub>15</sub>



# Exercice 3 (Solution)

```
Algorithme Tableau1D;
 Tableau A(50): Entier;
 Variable N, elt, i, j: Entier;
Début
 Ecrire ('Entrer la taille:');
 Lire (N);
 Ecrire ('Saisir le tableau:');
 pour i←1 à N pas 1
    Lire (T(i));
 finpour
```

```
Ecrire ('Afficher avant la suppressions:');

pour i←1 à N pas 1

Ecrire (T(i));

finpour

Ecrire ('Entrer elt à supprimer:');

Lire (elt);
```



# Exercice 3 (Solution)

```
pour i←1 à N pas 1
 si A(i) = elt alors
     pour j←i à N-1 pas 1
       A(j) \leftarrow A(j+1);
     finpour
    N←N-1;
    i←i-1;//Problème de adjacent
finsi
finpour
```

```
Ecrire ('Afficher après la suppression:');
 pour i←1 à N pas 1
    Ecrire (A(i));
 finpour
Fin
```



#### **Insertion:**

On considère les déclarations suivantes:

- tab: Nom du tableau

- N: la taille du tableau

- id: l'indice

- e : l'élément à insérer dans un tableau trié

tab: 6 7 9 10

**e**: 8

1- Rechercher

2- Décaler à droite

3- insérer



```
tab:
Recherche:
                      e:
                                                  6
                                                                 9
                                                                        10
id ← 1;
Tant que e>tab(id) et id<=N-1 faire
                                                       Position d'insertion: 3
id \leftarrow id+1;
fintantque
Décalage à droite:
pour j \leftarrow N \grave{a} id+1 pas -1
                                                  6
                                                                 9
                                                                        10
tab(j) \leftarrow tab(j-1);
fintantque
                                                  6
                                                                        9
                                                                               10
Insertion:
tab(id) \leftarrow e;
                                                                               10
                                                  6
                                                                        9
                                                                             19
```

Prof: R. EL AYACHI



#### Tri par sélection:

On considère les déclarations suivantes:

- tab: Nom du tableau

- N: la taille du tableau

- id: l'indice

- pmin: l'indice du minimum

- j: l'indice de recherche

- tmp: la permutation des contenus de deux

éléments

tab:

9 10 7 6 8	
------------	--

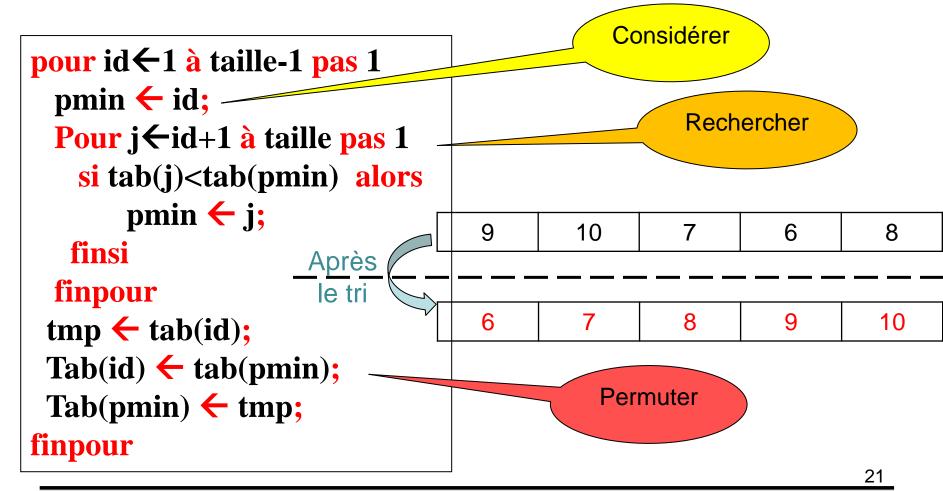
1- Considérer

2- Rechercher

3- Permuter



### Tri par sélection:



Prof: R. EL AYACHI

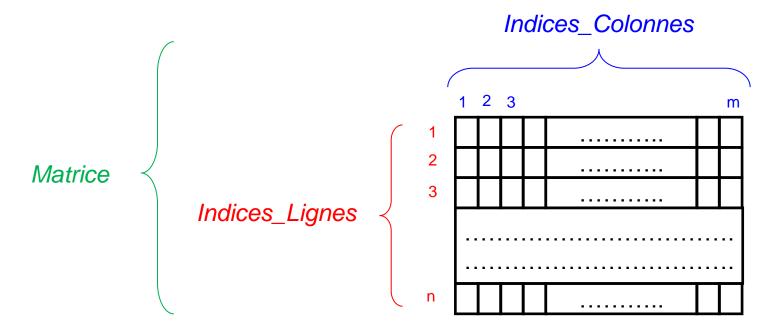
## **Tableaux**



# Tableau 2D



• Tableau à plusieurs indices (exemple: 2 indices)



- Identificateur du tableau: Matrice
- Contenu de l'élément (1,1): *Matrice*(1,1)

- Nombre de lignes : *n* 

- Contenu de l'élément (i,j): *Matrice*(i,j)

- Nombre de colonnes: *m* 

- . . .



#### **Déclaration:**

Tableau Nom\_tableau(lignes,colonnes):Type;

### **Exemple:**

**Tableau** Notes(5,4):**Entier**;

Tableau tab(10,20):Réel;



#### **Lecture:**

On considère les déclarations suivantes:

- mat: Nom du tableau
- lig: Nombre de lignes
- col: Nombre de colonnes
- i : L'indice ligne
- j : L'indice colonne

```
Ecrire('Saisir une matrice: ');

Pour i←1 à lig pas 1

Pour j←1 à col pas 1

Lire(mat(i,j));

finpour

finpour
```



### **Affichage:**

On considère les déclarations suivantes:

- mat: Nom du tableau
- lig: Nombre de lignes
- col: Nombre de colonnes
- i : L'indice ligne
- j : L'indice colonne

```
Ecrire('Afficher une matrice: ');

Pour i←1 à lig pas 1

Pour j←1 à col pas 1

Ecrire(mat(i,j));

finpour

finpour
```



## **Exercice 4**

Ecrire un algorithme qui permet de saisir et d'afficher les éléments d'un tableau 2D.



# Exercice 4 (Solution)

```
Algorithme Tableau2D;
  Tableau M(50,50): Entier;
  Variable lig, col, i, j: Entier;
Début
 Ecrire ('Entrer lig et col:');
 Lire (lig,col);
 Ecrire ('Saisir la matrice:');
 pour i \leftarrow 1 à lig pas 1
     pour j \leftarrow 1 à col pas 1
         Lire (M(i,j));
     finpour
 finpour
```

```
Ecrire ('Afficher le tableau:');

pour i←1 à lig pas 1

pour j←1 à col pas 1

Ecrire (M(i,j));

finpour

finpour

Fin
```



## **Exercice 5**

Ecrire un programme en langage C qui calcule la transposée d'une matrice.

1	1	1		1	2	3
2	2	2	<b></b>	1	2	3
3	3	3		1	2	3



# Exercice 5 (Solution)

```
Algorithme Transposée;
  Tableau M(50,50),T(50,50): Entier;
  Variable lig, col, i, j: Entier;
Début
 Ecrire ('Entrer lig et col:');
 Lire (lig,col);
 Ecrire ('Saisir la matrice:');
 pour i \leftarrow 1 à lig pas 1
     pour j \leftarrow 1 à col pas 1
         Lire (M(i,j));
     finpour
 finpour
```

```
Ecrire ('Afficher le tableau:');
  pour i \leftarrow 1 à lig pas 1
      pour j \leftarrow 1 à col pas 1
           Ecrire (M(i,j));
      finpour
 finpour
pour i \leftarrow 1 à lig pas 1
      pour j \leftarrow 1 à col pas 1
           T(j,i) \leftarrow M(i,j)
      finpour
 finpour
```



## Exercice 5 (Solution)

```
Ecrire ('Afficher la transposée:');

pour i←1 à lig pas 1

pour j←1 à col pas 1

Ecrire (T(i,j));

finpour

finpour

Fin
```