

ALGORITHMIQUE ET PROGRAMMATION INFORMATIQUE

Chapitre : les tableaux

I. Utilisés

Imaginons que dans un programme, on ait besoin de 12 valeurs (par exemple les notes d'un étudiant sur un semestre, pour calculer sa note moyenne).

Une façon consisterait à créer 12 variables et à les moyenner :

Moy $(N1+N2+N3+N4+N5+N6+N7+N8+N9+N10+N11+N12)/12$;

Imaginons si on avait besoin de 100 valeurs, allons-nous déclarer 100 variables ?

II. Définitions

Tableau

- Un **tableau** est une collection de données homogènes, accessibles par un indice entier.
- Un ensemble de valeurs de même type, portant le même nom de variable, stockées de manière contiguë et repérées par un nombre.
- Il est aussi appelé **variable indicée**

Indice

C'est le nombre qui, au sein d'un tableau, sert à repérer chaque valeur

Éléments du tableau

Les éléments d'un tableau ont les valeurs contenues dans chaque cellule (case) de ce tableau.

Chaque élément du tableau est désigné le nom du tableau, suivi de l'indice de l'élément, entre parenthèses.

I. Tableau à une dimension

Les tableaux monodimensionnels sont des tableaux qui ont une ligne et une colonne.

Exemple de tableau à 5 cases

--	--	--	--	--

1. Déclaration

- On définit un nom pour le tableau
- On définit à l'avance sa taille, en précisant son plus grand indice (c'est la convention assez répandue dans les langages informatiques)

- On définit ce qu'il va contenir (valeurs numériques, alphanumériques, booléennes)

i. Remarque

Selon les langages de programmation, les tableaux commencent:

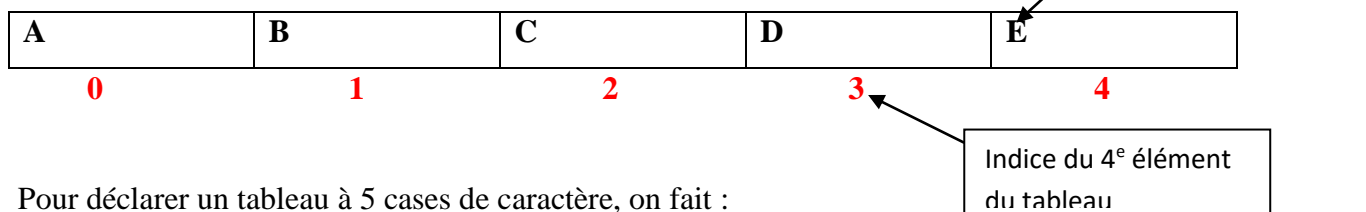
- à l'indice 1 (pseudo langages, Pascal)
- à l'indice 0 (en C et C++)

NB : nous utiliserons le cas où l'indice du tableau commence à zéro.

ii. Syntaxe

Tableau *nom_du_tableau*[taille] : *type*

Exemple :



Pour déclarer un tableau à 5 cases de caractère, on fait :

Tableau tab [4] : caractère

2. Accès à un élément du tableau

Ensuite, on peut accéder au i-ème élément du tableau en écrivant t[i] ou t(i).

Exemple : t [3] représente le 4^{ème} élément du tableau c'est-à-dire la lettre D dans l'exemple précédent.

3. Déclaration et Initialisation

Soit le tableau suivant :

10	4	6	8	3
----	---	---	---	---

Tableau Tab[] : entier ← {10,4,6,8,3}

ou

tableau Tab[5] : entier ← {10,4,6,8,3}

4. Lecture et écriture

Pour lire ou écrire un tableau a de N éléments, on utilisera des boucles:

- **Ecriture ou remplissage**
pour i de 0 à N-1 faire

```

        écrire ("entrez l'élément " i+1)
        lire(t[i])
    fin pour

```

Ou bien

```

    pour i de 1 à N faire
        écrire ("entrez l'élément " i)
        lire(t[i-1])
    fin pour

```

Exemple complet

```

Algorithme SaisieTableau
{Saisie tous les éléments d'un tableau}
tableau t : entier
i : entier
Début
    Pour i de 0 à tailleMax-1 faire
        écrire ("entrez l'élément " i+1)
        lire(t[i])
    Fin Pour
Fin

```

- **Parcours ou lecture d'un tableau**

```

    pour i de 0 a N-1 faire
        écrire(t[i])
    fin pour

```

ou encore

```

    pour i de 0 a N-1 faire
        écrire(t[i])
    fin pour

```

Exemple complet

```

Algorithme AfficheTableau
{Affiche tous les éléments d'un tableau}
tableau t : entier
i : entier
Début

```

```

    Pour i de 0 à tailleMax-1 faire
        Écrire(t[i])
    Fin Pour
Fin

```

Exemple complet des notes

Algorithme moyenne

Tableau Note(11) : réel;

Moy, Som : réel;

i en Entier;

Début

 Pour i de 0 à 10

 Ecrire (“Entrer la note ”, i+1) ;

 Lire Note(i) ;

 i Suivant

 Som ← 0 ;

 Pour i de 0 à 10

 Som ← Som + Note (i) ;

 i Suivant

 Moy ← Som / 12 ;

Fin

- **Recherche des plus petit et grand éléments d'un tableau**

```

Algorithme Maximum
{Recherche l'élément le plus grand d'un tableau de taille N non nulle }
tableau t : entier
i, max : entier
Début
    max ← t[1]
    Pour i ← 2 à N faire
        Si (t[i] > max) Alors
            max ← t[i]
        Fin si
    Fin Pour
    Écrire(max)
Fin

```

- **On cherche la position du minimum dans un tableau**

```

Algorithme PositionMinimum
{Recherche l'élément le plus petit d'un tableau de taille N non nulle et
affiche sa position c.-à-d. son indice}
tableau t : entier

```

```

d,f : entier
i,imin : entier
Début
    imin ← 0
    Pour i ← 0 à tailleMax-1 faire :
        Si t[i] ≤ t[imin] alors
            imin ← i
        Fin si
    Fin pour
    Ecrire(imin)
Fin

```

• Existence d'un élément dans un tableau

```

Algorithme Recherche
{Indique si l'élément e est présent ou non dans le tableau t de taille N }
e, i : entier
tableau t : entier
Début
    i ← 1;
    Tant que (i ≤ N) et (t[i] ≠ e) faire
        i ← i+1
    Fin tant que
    Si (i > N )
        Alors Ecrire("l'élément recherché n'est pas présent")
    Sinon
        Ecrire("l'élément recherché a été découvert")
    Fin si
Fin

```

5. Conventions

Dans un tableau, la valeur d'un indice doit toujours :

- **être égale au moins à 0.** Donc attention, Tab(6) est le septième élément du tableau Tab.
- **être un nombre entier.** Quel que soit le langage, l'élément Truc(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). Si le tableau Tab a été déclaré comme ayant 25 éléments, la présence dans une ligne, sous une forme ou sous une autre, de Tab(32) déclenchera automatiquement une erreur.

Remarque : Il ne faut pas confondre, dans sa tête et / ou dans un algorithme, l'indice d'un élément d'un tableau avec le contenu de cet élément ; c'est-à-dire i et Tab(i).

En notation algorithmique, il n'y a aucun rapport entre i et truc(i).

6. Tableaux dynamiques

- On ne connaît souvent pas à l'avance la taille nécessaire pour un tableau
- On pourrait toujours déclarer un tableau avec 10000 valeurs...
- Mais on n'est pas toujours sûr que ce serait suffisant et ça prend beaucoup de place pour rien si on a besoin de moins...
- Aussi, il est possible de déclarer un tableau sans spécifier sa taille. Puis, à l'intérieur du programme, on peut spécifier sa taille grâce à l'instruction nommée **Redim**.

Voici notre exemple sur la note moyenne d'un étudiant :

```

Tableau Notes[] : Entier;
Nb : Entier;
Début
    Ecrire ("Combien de notes à entrer ?") ;
    Lire Nb ;
    Redim Notes[Nb-1] ;
    ...
Fin

```

II. Tableau multi-dimensions

Les **tableaux à deux dimensions** sont des tableaux qui ont au moins 1 ligne et une colonne.

Ils sont utilisés pour résoudre certains problèmes : par exemple un jeu de damier, la somme de deux matrices, etc.

1. Déclaration

Syntaxe

Tableau *nom_du_tableau*[*nombreLigne*][*nombreColonne*] : *type*

Exemple :

0	A	B	C	D	E
1	Z	Y	X	W	V
	0	1	2	3	4

La valeur l'élément de la 1^{ère} ligne 5^{ème} colonne du tableau

Indice de l'élément de la 1^{ère} ligne 5^{ème}

Pour déclarer un tableau à 2 lignes et 5 colonnes (10 cases) de caractère, on fait :

Tableau tab [2] [4] : caractère

Remarque :

Tableau Cases[9][9] : Entier;

Cela signifie : "réserve-moi un espace mémoire de 10 x 10 entiers et quand j'aurai besoin de l'une de ces valeurs, je le localiserai par deux indices" (comme à la bataille navale ou excel, sauf qu'on a pas une lettre et un nombre mais deux nombres).

2. Accès à un élément du tableau

Ensuite, on peut accéder à l'élément à la i-ème ligne et j-ème colonne du tableau en écrivant `t[i][j]` ou `t(i)(j)`.

3. Déclaration et Initialisation

Soit le tableau suivant :

10	4	6	8	3
50	5	2	0	4.50

Tableau Tab[][] : entier $\leftarrow \{ \{ 10,4,6,8,3 \}, \{ 50,5,2,0,4.50 \} \}$

ou

tableau Tab[2][5] : entier $\leftarrow \{ \{ 10,4,6,8,3 \}, \{ 50,5,2,0,4.50 \} \}$

4. Lecture et écriture

Le parcours ou remplissage d'un tableau se fait toujours par ligne-colonne et non colonne-ligne.

Pour lire ou écrire un tableau a de L lignes et C colonnes c'est-à-dire L x C éléments, on utilisera des boucles:

- **Ecriture ou remplissage**

pour i de 0 à L-1 faire

 pour j de 0 à C-1 faire

 écrire ("entrez l'élément " i+1, j+1)

 lire(`t[i][j]`)

 finpour

fin pour

ou encore

```
pour i de 1 à L faire
    pour j de 1 à C faire
        ecrire ("entrez l'élément " i, j)
        lire(t[i-1][j-1])
    finpour
fin pour
```

Exemple complet

```
Algorithme SaisieTableau
{Saisie tous les éléments d'un tableau}
tableau t[][] : entier
i, j : entier
Début
    Pour i de 0 à nombreLigneMax-1 faire
        Pour j de 0 à nombreColonneMax-1 faire
            ecrire ("entrez l'élément " i+1, j+1)
            lire(t[i][j])
        Fin Pour
    Fin Pour
Fin
```

- **Parcours ou lecture d'un tableau à L lignes et C colonnes**

```
pour i de 0 à L-1 faire
    pour j de 0 à C-1 faire
        ecrire(t[i][j])
    finpour
finpour
```

Ou bien

```
pour i de 1 à L faire
    pour j de 1 à C faire
        ecrire(t[i-1][j-1])
    finpour
finpour
```

Exemple complet

```
Algorithme AfficheTableau
{Affiche tous les éléments d'un tableau à nbLigne lignes et nbColonne
colonnes connus à l'avance}
tableau t[nbLigne][ nbColonne] : entier
```

```

i,j : entier
Début
  Pour i de 0 à nbLigne-1 faire
    Pour j de 0 à nbColonne-1 faire
      Écrire(t[i][j])
    Fin Pour
  Fin Pour
Fin

```

- **Recherche des plus petit et grand éléments d'un tableau**

```

Algorithme Maximum
{Recherche l'élément le plus grand d'un tableau de taille LxC non nulle }
tableau t[][] : entier
i, j, max : entier
Début
  max ← t[0][0]
  Pour i ← 2 à L-1 faire
    Pour j ← 1 à C-1 faire
      Si (t[i][j] > max) Alors
        max ← t[i][j]
      Fin si
    Fin Pour
  Fin Pour
  Écrire(max)
Fin

```

- **On cherche la position du minimum dans un tableau**

```

Algorithme PositionMinimum
{Recherche l'élément le plus petit d'un tableau de taille LxC non nulle et
affiche sa position c.-à-d. son indice}
tableau t[][] : entier
i, j, imin,jmin : entier
Début
  imin ← 0
  jmin ← 0
  Pour i ← 0 à L-1 faire :
    Pour j ← 0 à C-1 faire :
      Si t[i][j] ≤ t[imin][jmin] alors
        imin ← i
        jmin ← j
      Fin si
    Fin pour
  Fin pour
  Écrire(imin "et" jmin)
Fin

```

TD 5

Exercice 0

- Ecrire un algorithme permettant de déclarer et de remplir un tableau de 7 valeurs numériques de type entier en les mettant toutes à zéro.
- Ecrire un algorithme permettant de déclarer et de remplir un tableau de 7 valeurs numériques de type entier en les mettant toutes à zéro.

Exercice 1

Ecrire un algorithme qui déclare et remplit un tableau de 7 valeurs numériques en les mettant toutes à zéro.

Exercice 2

Ecrire un algorithme qui déclare et remplit un tableau contenant les six voyelles de l'alphabet latin.

Exercice 3

Ecrire un algorithme qui déclare un tableau de 9 notes, dont on fait ensuite saisir les valeurs par l'utilisateur.

Exercice 4

Que produit l'algorithme suivant ?

Tableau Nb(5) : Entier;

i : Entier;

Début

 Pour i allant de 0 à 5

 Nb(i) ← i * i ;

 i Suivant

 Pour i allant de 0 à 5

 Ecrire Nb(i) ;

 i Suivant

Fin

Peut-on simplifier cet algorithme avec le même résultat ?

Exercice 5

Ecrivez la fin de l'algorithme 3 afin que le calcul de la moyenne des notes soit effectué et affiché à l'écran.

Exercice 6

Ecrivez un algorithme permettant à l'utilisateur de saisir un nombre quelconque de valeurs, qui devront être stockées dans un tableau.

L'utilisateur doit donc commencer par entrer le nombre de valeurs qu'il compte saisir. Il effectuera ensuite cette saisie. Enfin, une fois la saisie terminée, le programme affichera le nombre de valeurs négatives et le nombre de valeurs positives.

Exercice 7

Ecrivez un algorithme calculant la somme des valeurs d'un tableau (on suppose que le tableau a été préalablement saisi).

Exercice 8

Ecrivez un algorithme constituant un tableau, à partir de deux tableaux de même longueur préalablement saisis. Le nouveau tableau sera la somme des éléments des deux tableaux de départ. Par exemple :

Tableau 1 :

Tableau 2 :

Tableau à constituer :

1	2	3	4
5	12	-2	0
6	14	1	4

Exercice 9

Toujours à partir de deux tableaux précédemment saisis, écrivez un algorithme qui calcule le schtroumpf des deux tableaux. Pour calculer le schtroumpf, il faut multiplier chaque élément du tableau 1 par chaque élément du tableau 2, et additionner le tout. Par exemple si l'on a :

Tableau 1 :

4	8	7	12
---	---	---	----

 et Tableau 2 :

3	6
---	---

Le Schtroumpf sera :

$$3*4 + 3*8 + 3*7 + 3*12 + 6*4 + 6*8 + 6*7 + 6*12 = 279$$

Exercice 10

Ecrivez un algorithme qui permette la saisie d'un nombre quelconque de valeurs, sur le principe de l'exercice 7. Toutes les valeurs doivent être ensuite augmentées de 1, et le nouveau tableau sera affiché à l'écran.

Exercice 11

Ecrivez un algorithme permettant, toujours sur le même principe, à l'utilisateur de saisir un nombre déterminé de valeurs. Le programme, une fois la saisie terminée, renvoie la plus grande valeur en précisant quelle position elle occupe dans le tableau. On prendra soin d'effectuer la saisie dans un premier temps, et la recherche de la plus grande valeur du tableau dans un second temps.

Exercice 12

Toujours et encore sur le même principe, écrivez un algorithme permettant, à l'utilisateur de saisir les notes d'une classe. Le programme, une fois la saisie terminée, renvoie le nombre de ces notes supérieures à la moyenne de la classe.

Exercice 13

Ecrivez un algorithme qui permette la saisie d'un polynôme d'un degré quelconque (qui sera demandé à l'utilisateur). On rappelle qu'un polynôme de degré N est de la forme :

$$P(X) = a_0 + a_1 \cdot X + a_2 \cdot X^2 + \dots + a_N \cdot X^N$$

Exercice 14

Ecrivez un algorithme qui permette la saisie de deux polynômes de degré quelconque (et donc pas nécessairement les mêmes) et qui en calcule leur somme. Par exemple, si les polynômes saisis sont :

$$P1(X) = 1 + X + X^2 \text{ et } P2(X) = 3 + 2 \cdot X^2 - 7 \cdot X^3, \text{ le polynôme résultant sera } P3(X) = 4 + X + 3 \cdot X^2 - 7 \cdot X^3.$$

Exercice 15

Ecrivez un algorithme qui permette la saisie de deux polynômes de degré quelconque (et donc pas nécessairement les mêmes) et qui en calcule leur produit. Par exemple, si les polynômes saisis sont :

$$P1(X) = 1 + X + X^2 \text{ et } P2(X) = 3 + 2 \cdot X^2, \text{ le polynôme résultant sera } P4(X) = 3 + 3 \cdot X + 5 \cdot X^2 + 2 \cdot X^3 + 2 \cdot X^4.$$

Exercice 16

Ecrivez un algorithme remplissant un tableau de 6 sur 13 valeurs, avec des zéros.

Exercice 17

Que produiront ces algorithmes ?

<pre>Tableau X(1,2) : Entier; Variables i, j, val : Entier; Début val 1 Pour i 0 à 1 Pour j 0 à 2 X(i,j) ← val val ← val + 1 j Suivant i Suivant Pour i de 0 à 1 Pour j de 0 à 2 Ecrire(X(i,j)) j Suivant i Suivant Fin</pre>	<pre>Tableau X(1,2) : Entier; Variables i, j, val : Entier; Début val ← 1 ; Pour i de 0 à 1 Pour j 0 à 2 X(i, j) val ; val ← val + 1 ; j Suivant i Suivant Pour j de 0 à 2 Pour i de 0 à 1 Ecrire (X(i,j)); i Suivant j Suivant Fin</pre>
---	---