

Algorithmme et programmation

Chapitre 3 : Les tableaux

3.1) Les tableaux

Un tableau est un ensemble de données de même type.

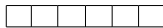
Exemple de problème : afficher une suite de nombres après avoir divisé tous les nombres par le maximum de ces nombres.

Cela nécessite de **conserver en mémoire** tous les nombres.

Variable simple : une valeur.



Variable de type tableau : une collection de valeurs de même type.



Indice (=adresse de chaque case).

Donc un tableau est un ensemble de valeurs dont chaque élément est indicé (il y a un indice) et est de même type.

Déclaration :

var

NomTableau : tableau de Type [d..f]

d..f représente l'intervalle des valeurs (entières) que peut prendre l'indice.

- Si i est un indice, alors $i \in [d..f]$
- La taille du tableau (nombre d'éléments) vaut : $f - d + 1$
- Les bornes d et f sont soit des entiers, soit des constantes (type entier).

Exemple de déclaration :

var

MonTab : tableau d'entier [0..5]

0	1	2	3	4	5
5	8	1	-4	7	0

T2 : tableau de caractères [1..4]

1	2	3	4

Accès et utilisation :

Chaque case peut-être considérée comme une variable.

- Accès en lecture :

écrire (t2[3]) /* Affiche la valeur de la case d'indice 3 du tableau t2*/

Soit v une variable de type entier :

$v \leftarrow \text{MonTab}[4]$ /* v vaudra 7*/

- Accès en écriture :

MonTab[0] \leftarrow -4

0	1	2	3	4	5
5	8	1	-4	7	0



0	1	2	3	4	5
-4	8	1	-4	7	0

Il existe deux façons pour rentrer une valeur dans le tableau :

écrire (« Saisir un entier »)
lire (MonTab[0])



écrire (« Saisir un entier »)
lire (x)
MonTab [0] ← x

ATTENTION : Interdiction de faire « MonTab ← 4 », les cases du tableau ne vont pas toutes prendre la valeur 4.

3.2) Parcours d'un tableau

Parcours total :

Pour parcourir totalement un tableau, on utilise des boucles inconditionnelles.

Exemple :

1) Remplir un tableau.

Algo SaisieTab

```
const
    TAILLE = 5
var
    t : tableau d'entiers [1..TAILLE]
    i : entier
debut
    /*On donne une valeur à chaque élément du tableau*/
    pour i de 1 à TAILLE faire
        écrire (« entrez le », i, « entier »)
        lire (t[i])
    finPour
fin
```

2) Multiplier tous les éléments par 2

Algo Multiplie2

```
const
    TAILLE = 5
var
    t : tableau d'entiers [1..TAILLE]
    i : entier
debut
    pour i de 1 à TAILLE faire
        t[i] ← t[i]*2
    finPour
fin
```

Parcours partiel :

Pour parcourir partiellement un tableau, on utilise des boucles conditionnelles.

Exemple :

Tester si les éléments de t sont tous positifs.

Algo TestPositif

```
const
    TAILLE = 5
var
    t : tableau d'entiers [1..TAILLE]
    i : entier
    estPos : booléen
debut
    pour i de 1 à TAILLE faire
        ecrire (« entrez le », i, « entier »)
        lire (t[i])
    finPour
    i ← 1
    estPos ← vrai
    tantque (i ≤ TAILLE et estPos) faire
        si t[i] < 0
            alors estPos ← FAUX
        finSi
    finTantQue
    si estPos
        alors ecrire (« Le tableau ne contient que des positifs »)
        sinon ecrire (« Le tableau a des valeurs négatives »)
    finSi
fin
```

ATTENTION : Un tableau de n cases ne contient pas forcément n éléments.

Exemple :

Saisie d'entier avec drapeau.

Algo Saisie

```
const
    MAX = 1000
    DRAPEAU : '\'
var
    t : tableau de caractères [1..MAX-1]
    i : entier
    varLu : caractère
debut
    i ← 0
    répéter
        ecrire (« Entrez un caractère ou », DRAPEAU, « pour arrêter »)
        lire (varLu)
        t[i] ← varLu
        i++
    tantque (varLu ≠ DRAPEAU et i < MAX)
fin
```

Ne pas demander les cases qui n'ont pas été saisies.

ATTENTION AU DEBORDEMENT :

Si le tableau est partiellement rempli, pour le parcourir il faut connaître le nombre d'éléments insérés. Pour le parcourir on utilise une boucle pour (« Pour i de 1 à nb faire ... » avec $nb \leq MAX$).

3.3) Tableaux à plusieurs dimensions

Tableaux à deux dimensions :

Un tableau à deux dimensions est comme une grille a n lignes et m colonnes.

Déclaration :

var

NomTableau : tableau de type [d1..f1][d2..f2]

↗ ↖
Lignes Colonnes

[d1..f1] }
[d2..f2] } Intervalle des indices de
chaque dimension

Exemple :

var

mat : tableau de réels [0..2][1..6]

2						
1						
0						
	1	2	3	4	5	6

Accès direct à cette case : mat [1][2]
Située sur la ligne d'indice 1 et la colonne d'indice 2

Accès au tableau :

t[i][j]

avec i l'indice de la ligne et j l'indice de la colonne.

Parcours complet :

On utilise deux boucles pour imbriquées.

Exemple :

Algo Parcours2D

const

nbligne = 100

nbcolonne = 100

var

mat : tableau de réels [1..nbligne][1..nbcolonne]

i, j : entier

debut

pour i de 1 à nbligne faire

pour j de 1 à nbcolonne faire

ecrire (« Ligne »,i « colonne »,j)

lire (mat[i][j])

finpour

finpour

fin

} Pour un i donné, je parcours tous les
j

I	J	Lire (mat[i][j])
1	1	lire(mat[1][1])
1	2	lire(mat[1][2])
1	...	lire(mat[1][...])
1	nbcolonne	lire(mat[1][nbcolonne])
2	1	lire(mat[2][1])
2	...	lire(mat[2][...])
2	nbcolonne	lire(mat[2][nbcolonne])
...	...	lire(mat[...][...])
nbligne	1	lire(mat[nbligne][1])
nbligne	...	lire(mat[nbligne][...])
nbligne	nbcolonne	lire(mat[nbligne][nbcolonne])