

Les tableaux

Ensemble de données du même type

Exemple de problème :

Saisir une suite de nombres, puis afficher cette suite après avoir divisé tous les nombres par la valeur maximale de la suite.

Nécessité de conserver les nombres en mémoire

variable contenant une valeur

12

variable contenant une collection de valeurs du même type

tabVal contenant des les valeurs val

12 20 -30 1 2 56 -5

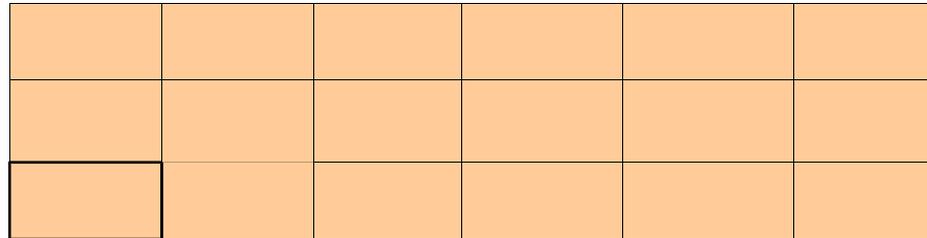
Les tableaux

Structure de données permettant d'effectuer un même traitement sur des données de même nature.

tableau à **une**
dimension



tableau à **deux**
dimensions



Exemples d'applications

- Ensemble de valeurs entières,réelles, booléennes,....
- Ensemble de noms (type chaîne)
- Ensemble de caractères (type caractère)
- Ensemble d'adresses (type Adresse:nom, adresse, num téléphone)
- Ensemble d'ouvrages

Traitements sur les tableaux

- On veut pouvoir :
 - **créer** des tableaux
 - **ranger** des valeurs dans un tableau
 - **récupérer, consulter** des valeurs rangées dans un tableau
 - **rechercher** si une valeur est dans un tableau
 - **mettre à jour** des valeurs dans un tableau
 - **modifier** la façon dont les valeurs sont rangées dans un tableau (par exemple : les trier de différentes manières)
 - effectuer des **opérations entre tableaux** : comparaison de tableaux, multiplication,...
 - ...

Définition du type

nom du
tableau

T1

1	2	3	4	5	6
23	11	-1	8	-29	15

indice
du tableau

contenu
du tableau

T2

1	2	3	4	5	6	7
t	a	b	l	e	a	u

Remarques :

- 1) Indices : en général, démarrage à 1, **mais en C, démarrage à 0**
- 2) Nombre d'octets occupés : dépend du type des valeurs enregistrées

Déclaration d'un tableau

Exemple : déclaration d'un tableau pouvant contenir jusqu'à 35 entiers

```
variable tab : tableau [1, 35] d'entiers
```

nom du tableau

mot clé

indices min et max
(taille)

type des éléments

Définition d'un type tableau

Exemple : déclaration d'un tableau type chaine contenant 10 caractères

```
type chaine = tableau [1,10] de caractères
```

Exemple : utilisation du type chaine pour déclarer des variables contenant 10 caractères

```
variables nom, prénom : chaine
```

Utilisation d'un tableau: les indices

- Accès en écriture :
 - Écrire(tab[2]) *{le contenu du tableau à l'indice 2 est affiché à l'écran}*
- Accès en lecture :
 - tab[4] ← 1 *{la valeur 1 est placée dans le tableau à l'indice 4}*
 - Lire (tab[3]) *{la valeur entrée par l'utilisateur est enregistrée dans le tableau à l'indice 3}*
 - **Attention** : ~~tab ← 18~~ ~~nom[2] ← 3~~

Tableaux à deux dimensions

	1	2	3	4	5	6	7
1	1	3	5	7	9	11	13
2	0	2	4	6	8	10	12
3	11	25	29	36	27	24	20

tableau à 3 lignes et 7 colonnes

- **Déclaration:** matrice: tableau[1,3][1,7] d'entiers

indices min et max
des lignes

indices min et max
des colonnes

Tableaux à deux dimensions (suite)

	1	2	3	4	5	6	7
1	1	3	5	7	9	11	13
2	0	2	4	6	8	10	12
3	11	25	29	36	27	24	20

- **Accès en écriture :**

- Écrire(matrice[2,4]) {la valeur contenue en ligne 2 colonne 4 est affichée à l'écran}

- **Accès en lecture :**

- Lire (matrice[1,7]) {la valeur fournie est enregistrée en ligne 1 colonne 7}
- **Matrice[1,1] ← 20**

Saisir les valeurs d'un tableau 1D

```
Algorithme SaisieTableau // {remplit un tableau avec nb valeurs entières}
constantes (TailleMAX : entier) ← 100
variables nb, i : entier
           tab1D : tableau [1, TailleMAX] d'entiers
début
  Écrire("Combien de valeurs sont à saisir?")
  Lire (nb)
  si (nb > TailleMAX) alors // {saisie impossible : la capacité du tableau est dépassée}
    Écrire ("trop de valeurs à saisir")
  sinon
    pour i ← 1 à nb faire
      Écrire ("Donner une valeur")
      // {valeur à ranger dans la i-ème case du tableau}
      Lire (tab1D[i])
    fpour
  fsi
fin
```

Saisie avec « drapeau »

Algorithme SaisieTableauAvecDrapeau

// {remplit un tableau tant qu'il y a des caractères à ranger, arrêt dès la saisie d'un '.'}

constantes (TailleMAX : entier) ← 100

(Drapeau:caractère) ← '.'

variables i : entier

tab1D : tableau [1, TailleMAX] de caractères

Car : caractère

début

Écrire("Tapez un caractère, ou ", DRAPEAU, "pour arrêter la saisie. ")

Lire (Car)

i ← 0

tant que (Car <> DRAPEAU **et** i < TailleMAX) **faire**

i ← i + 1

tab1D[i] ← Car

Écrire (" Tapez un autre caractère, ou ", DRAPEAU, "pour arrêter la saisie.")

Lire (Car)

Ftq

si (Car == DRAPEAU)

alors écrire ("Valeurs saisies intégralement.")

sinon écrire ("Trop de caractères à saisir, plus de place ! ")

Fsi

fin

Affichage d'un tableau

Algorithme affichageTableau

constantes (TailleMAX : entier) ← 100

(Drapeau:caractère ← '.')

variables i,j : entier

tab : tableau [1, TailleMAX] de caractères

Car : caractère

début

Écrire("Tapez un caractère, ou ", DRAPEAU, "pour arrêter la saisie. ")

Lire (Car)

i ← 0

tant que (Car <> DRAPEAU **et** i < TailleMAX) **faire**

i ← i + 1

tab1D[i] ← Car

Écrire (" Tapez un autre caractère, ou ", DRAPEAU, "pour arrêter la saisie.")

Lire (Car)

Ftq

Pour j ← 1 à i **faire**

Écrire (tab1D[j])

Fpour

fin

Partie affichage



Lecture d'un tableau 2D

Algorithme SaisieTableau2D

constantes (TailleMAX : entier) ← 100

variables nbLignes, nbColonnes, idxL, idxC : entiers

tab2D : tableau [1, TailleMAX ; 1, TailleMAX] d'entiers

Début

Écrire ("saisissez le nombre lignes") ;

Lire (nbLignes)

Écrire ("saisissez le nombre colonnes") ;

Lire (nbColonnes)

si (nbLignes > TailleMAX) ou (nbColonnes > TailleMAX)

alors écrire ("trop de valeurs à saisir")

sinon

pour idxL ← 1 à nbLignes faire

pour idxC ← 1 à nbColonnes faire

Écrire ("Ligne" , idxL, "colonne", idxC, " : ")

Lire (**tab2D[idxL,idxC]**)

fpour

fpour

fsi

fin