

Langage C : tableaux

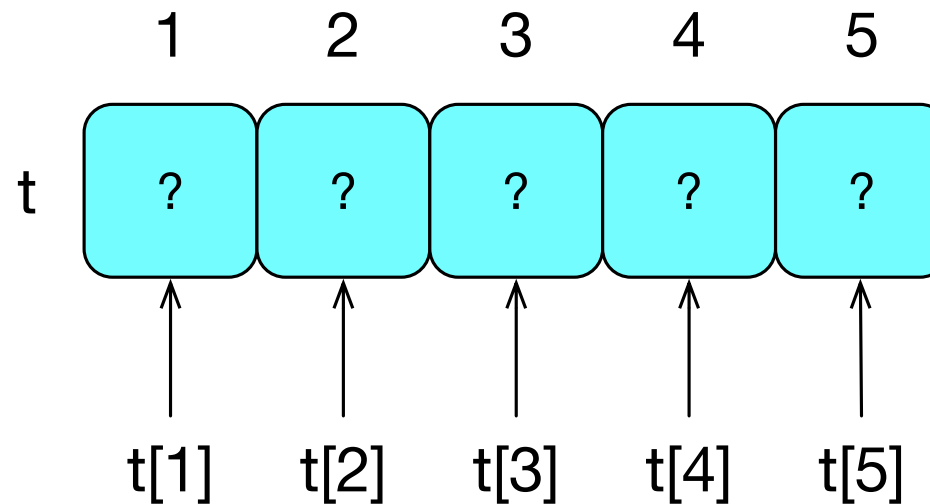
Sébastien Jean

IUT de Valence
Département Informatique

v1.0, 21 octobre 2025

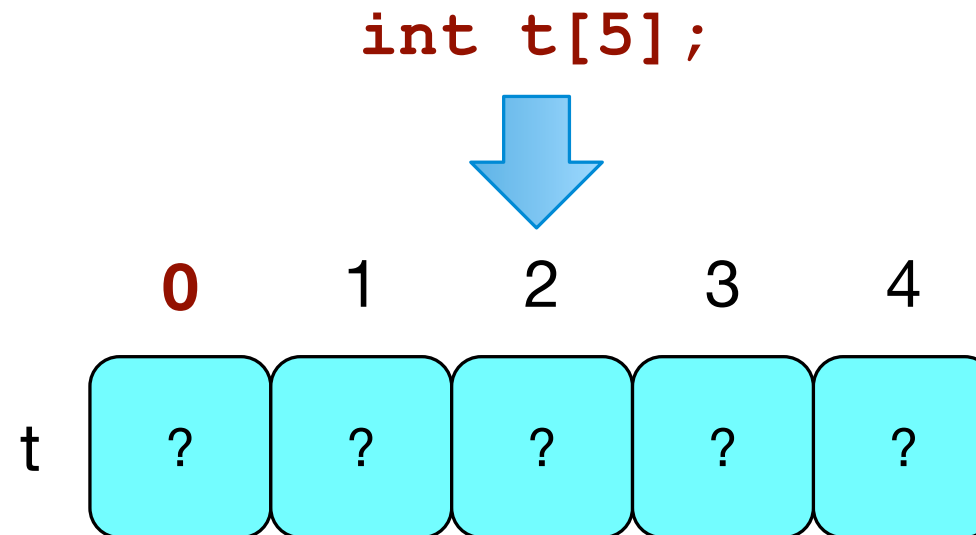
Tableaux : *Pseudo-code* (rappel)

VARIABLE t : tableau d'entiers [5]



Tableaux 1D en C : variables

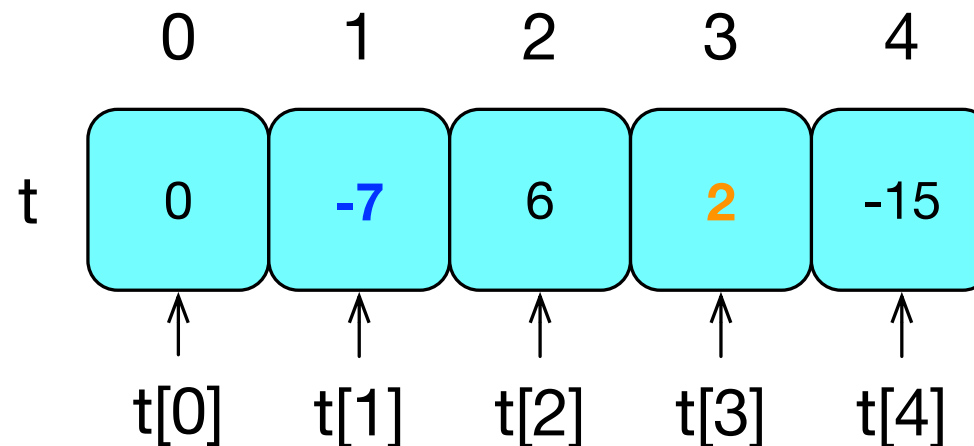
- **Déclaration** d'une variable suivant la syntaxe `type nom[taille]`



- La déclaration fait exister le tableau (et ses cases)
- Les **indices valides** sont sur l'**intervalle** `[0, taille-1]`

Tableaux 1D en C : lecture/affectation

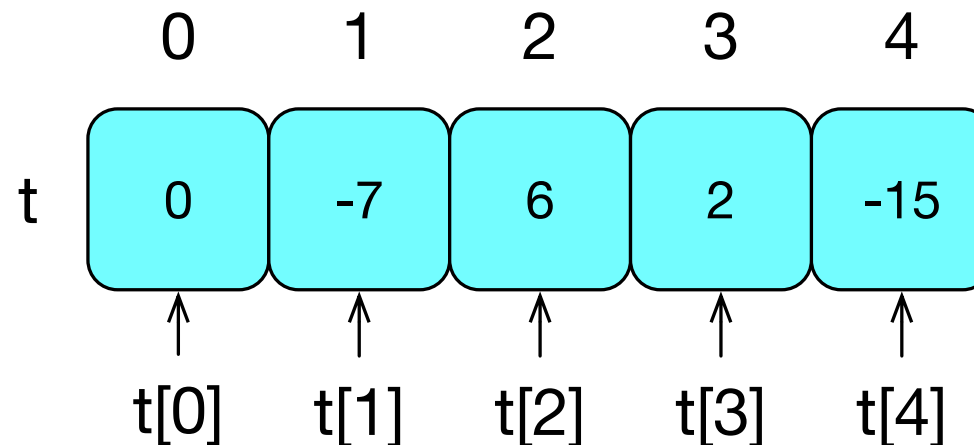
- Désignation d'une case suivant la syntaxe `nom[indice]`



- L'expression `t[1]` vaut le contenu de la 2e case (ici -7)
- L'affectation de 2 à la 4e case s'effectue via `t[3] = 2`

Tableaux 1D en C : initialisation

- Initialisation après déclaration
 - case par case, avec ou sans boucle for (indice de 0 à taille-1)
- Initialisation pendant la déclaration
 - Liste entre accolades de toutes les valeurs, dans l'ordre
 - Ex. : `int tab[3] = {3, 2, 1};`



Passage de tableau 1D en paramètre

- Paramètre de type tableau 1D dans la signature d'une fonction
→ **pas d'indication obligatoire de la taille**
- Appel de fonction avec un paramètre de type tableau
→ **variable de type tableau** passée comme **valeur**
 - N.B. : La **réaffectation des cases d'un tableau est permanente et visible au retour de la fonction** (cf. plus tard)

Code C

```
int f(int t[]) {  
    ...  
}  
  
int main() {  
    int tab[3] = {3, 2, 1};  
    printf("%d\n", f(tab));  
}
```

Interlude : Création/clonage d'un projet Gitlab



- **Créer un projet TableauxC** sur Gitlab (avec un README)
- **Cloner** le projet depuis *VsCode* et **ouvrir le dépôt**
- **Modifier** le fichier README.md pour indiquer à quoi sert ce projet
- N.B. : pour chacun des exercices ExerciceX suivant :
 - **Ecrire le programme** dans ExerciceX/src/main.c et le compiler dans ExerciceX/build/ExerciceX
 - Rédiger un jeu d'essai dans un fichier ExerciceX/Essai

Exercice 1 : spécification de la fonction

Enoncé du problème

Etant donnés un **tableau d'entiers** t et sa **taille** n , on souhaite calculer la **moyenne des éléments du tableau**.

Spécification du problème

- Donnée d'entrée : t , **tableau d'entiers** (tableau rempli de valeurs)
- Donnée d'entrée : n , **entier** (nombre de valeurs de t)
- Donnée de sortie : moy , **réel** (moyenne des éléments de t)
- Pré-condition : $n \geq 1$
- Post-condition : moy est égal à la valeur moyenne des éléments de t .

Signature de la fonction

- **moy_tab** (t : tableau d'entiers, n : entier) : **réel**

Exercice 1 : *pseudo-code*

```
FONCTION moy_tab (t : tableau d entiers,  n : entier)
                  : réel

VARIABLE somme    : entier
VARIABLE indice   : entier

somme ← 0

POUR indice de 1 A n PAR PAS DE 1

    somme ← somme + t[indice]

FIN POUR

RETOURNER somme / n

FIN FONCTION
```



Exercice 1 : traduction en C

Exemple de jeu d'essai

$t = \{-7, 5, 4, 0, -12\},$	$n = 5$	$\rightarrow -2.000000$
$t = \{1\},$	$n = 1$	$\rightarrow 1.000000$
$t = \{1, 2, 3\},$	$n = 3$	$\rightarrow 2.000000$
$t = \{0, 0, 0\},$	$n = 3$	$\rightarrow 0.000000$

Exercice 2 : spécification

Enoncé du problème

Etant donnés un **tableau d'entiers** t et sa **taille** n , on souhaite **afficher une représentation du contenu du tableau** sous la forme $\{v_1, v_2, \dots, v_n\}$.

Spécification du problème

- Donnée d'entrée : t , **tableau d'entiers** (tableau rempli de valeurs)
- Donnée d'entrée : n , **entier** (nombre de valeurs de t)
- Donnée de sortie : **aucune** (affichage seulement)
- Pré-condition : $n \geq 1$
- Post-condition : l'affiche produit est de la forme $\{v_1, \dots, v_n\}$, avec v_i représentant la valeur de la case d'indice i de t (pour i de 1 à n)

Signature de la fonction

- **affichage_tab** (t : **tableau d'entiers**, n : **entier**) : (**aucun**)

Exercice 2 : *pseudo-code*

```
FONCTION affichage_tab (t : tableau d'entiers ,  
                        n : entier) : (aucun)
```

```
VARIABLE indice: entier
```

```
afficher("{")
```

```
POUR indice de 1 A n PAR PAS DE 1
```

```
    afficher(t[indice])
```

```
    SI indice < n ALORS
```

```
        afficher(", ")
```

```
    FIN SI
```

```
FIN POUR
```

```
afficher("}")
```

```
afficher_saut_ligne()
```

```
FIN FONCTION
```



Exercice 2 : traduction en C

Exemple de jeu d'essai

$\{-7, 5, 4, 0, -12\}$ $\rightarrow$ $\{-7, 5, 4, 0, -12\}$
 $\{1\}$ $\rightarrow$ $\{1\}$

Exercice 3 : spécification

Enoncé du problème

Etant donnés un **tableau d'entiers** t , sa **taille** n et une **valeur** v , on souhaite **trouver l'indice de la première occurrence de v dans t si elle existe.**

Spécification du problème

- Donnée d'entrée : t , **tableau d'entiers** (tableau rempli de valeurs)
- Donnée d'entrée : n , **entier** (nombre de valeurs de t)
- Donnée d'entrée : v , **entier** (valeur à trouver)
- Donnée de sortie : **indice**, **entier** (l'indice recherché)
- Pré-condition : $n \geq 1$
- Post-condition : $1 \leq \text{indice} \leq n$ si la case d'indice indice de t vaut v et qu'aucune autre case d'indice inférieur à indice ne vaut v , $\text{indice} = -1$ si aucune case de t ne contient la valeur v .

Signature de la fonction

- **premier_indice** (t : **tableau d'entiers**, n : **entier**, v : **entier**) : **entier**

Exercice 3 : *pseudo code*

```
FONCTION premier_indice (t : tableau d'entiers,  
                        n : entier, v : entier) : entier  
  
    VARIABLE indice    : entier  
  
    POUR indice de 1 A n PAR PAS DE 1  
  
        SI t[indice] = v ALORS  
            RETOURNER indice  
  
    FIN POUR  
  
    RETOURNER -1  
  
FIN FONCTION
```



Exercice 3 : traduction en C

Exemple de jeu d'essai

$t = \{-7, 5, 4, 4, -12\},$	$n = 5, v = -7$	$\rightarrow 0$
$t = \{-7, 5, 4, 4, -12\},$	$n = 5, v = 4$	$\rightarrow 2$
$t = \{-7, 5, 4, 4, -12\},$	$n = 5, v = -12$	$\rightarrow 4$
$t = \{-7, 5, 4, 4, -12\},$	$n = 5, v = 1$	$\rightarrow -1$
$t = \{1\},$	$n = 5, v = 1$	$\rightarrow 0$
$t = \{1\},$	$n = 5, v = 0$	$\rightarrow -1$

Fin !

