

Tableau 1D : `int8_t t[N];`

N cases de type `int8_t`.

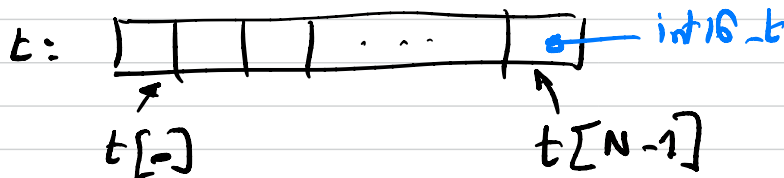
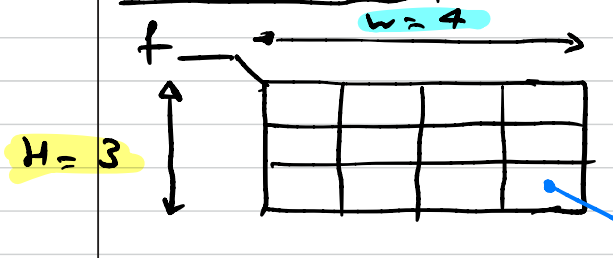


image ←

Tableau 2D :



nbre de lignes : $H=3$

nbre de cols : $W=4$

`int8_t f[H][W];`
 nbre de lignes nbre de colonnes

#define H 3
 #define W 4

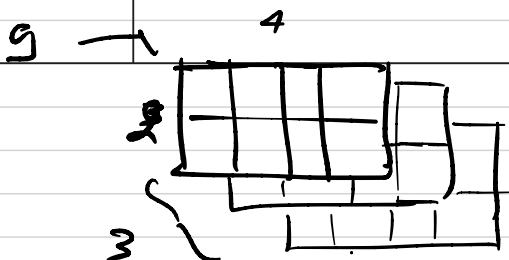
init tableau complet à zéro

`int8_t f[H][W] = {0};`

film ←

Tableau 3D

3 tableaux de 2 lignes x 4 colonnes



`double g[3][2][4] = {0};`

```
uint8_t f[H][W];
```

↑ ↑
nbre de nbre de
lignes colonnes

Exercice: TD 2021/01/11

main

- ① Créer un tableau 2D à `float` uint8_t
3 lignes, 4 colonnes
- ② Remplir toutes les cellules de ce tableau
avec des nombres aléatoires
entre [0 et 100] [0..99]
- ③ Afficher le contenu du tableau

13h55

Structurer le code avec:

1) une fonction pour l'initialisation de
tableau

2) une fonction pour l'affichage du tableau

```
void display ( uint8_t t[H][W], const uint32_t n );
```

1D uint8_t t[]
3D t[H][W]

3) appeler ces 2 fonctions de puis le main..

14h25

26	95	77	35
2	14	30	17
49	68	59	69

2	14	17	26
30	35	49	59
68	69	77	95

Voici mon tableau 2D !

"Voir" le Tableau 2D comme un tableau 1D

"t"

"r"

l=2
c=3
c=2
c=1
c=0
l=1
c=2
c=1
c=0
l=0
c=3
c=2
c=1
c=0

k=11
:
:
:
:
k=5
k=4
k=3
k=2
k=1
k=0

69
59
68
49
17
30
14
2
35
77
95
26

uint8_t t[H][W];

m nombre de cellule.

uint8_t r[H*W];

copie "à la main" de t → r.

k=

		0			1		2	
e	0	0	0	0	1	1	1	1
c	0	1	2	3	0	1	2	3
k	0	1	2	3	4	5	6	7

$$k = c + l * w$$

Pour le P.S.

- ① Remplir "r" avec les valeurs de "t"
- ② trier "r"
- ③ remettre "r" → "t" en 2D
- ④ afficher "t"