3 - LANGAGE C

LES TABLEAUX (STATIQUES)

Loïc Cuvillon
l.cuvillon@unistra.fr

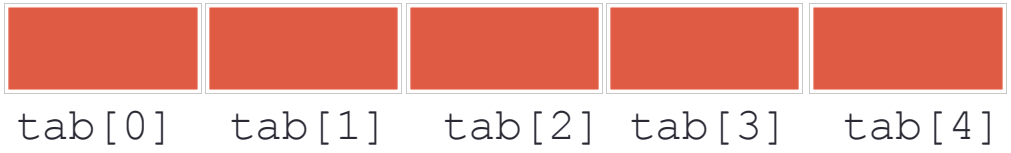
Sommaire chapitre 3

Définitions et déclarations de tableaux

Un tableau est une suite d'éléments de même type, stockés de façon contiguë en mémoire.

- Déclaration : `type nom[taille]`

```
char  ligne[80] ;
int   tab[5] ;
```



- Chaque élément est repéré par un indice (de 0 à taille-1) !
- On accède à un élément avec `tab[indice]`

Accès aux éléments

- `int t[5] ;`

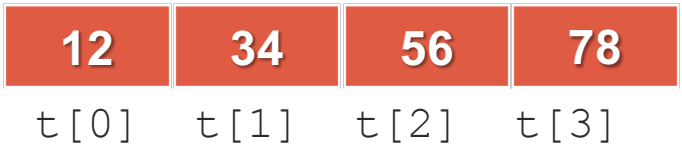
- `t[0] = 23 ;`

- `t[3] = t[0] + 4 ;`

- `i=2;`
  `t[i] = t[i+1] + t[0];`

Initialisation d'un tableau

- `int t[4] = {12,34,56,78};`



- `int t[4] = {1,2};`
  (éléments non spécifiés mis à 0)

- `int t[] = {3,2,1} ;`

  (taille définie par le nombre d'éléments)

- `int x=23;`

  ~~`int t[x];`~~ ← erreur !!!

Seule une constante est autorisée pour la taille. Pas de variable. (C ANSI)
On parle de tableau statique, la taille est fixe pendant toute l'exécution !

Manipuler un tableau

- initialiser, saisir, afficher, recopier un tableau : faire une boucle

```
int tableau[23];
int  i=0 ;

for (i=0; i<23; i++)
{
     tableau[i] = 0 ;
}

for (i=0; i<23; i++)
{
     scanf(”%d”,&tableau[i]);
}
```

Manipuler un tableau

- bien mieux avec un `#define` (lisibilité, mise à jour aisée de la taille) :

```
#define MAX_SIZE 23

int tableau[MAX_SIZE], i=0 ;

 for (i=0; i<MAX_SIZE; i++)
 {
      scanf("%d",&tableau[i]);
 }


 for (i=0; i<MAX_SIZE; i++)
 {
      printf("%d ",tableau[i]);
 }
```

`#define` permet de définir une constante `MAX_SIZE` et sa valeur (`23`) ne peut être modifiée par la suite
→ ce n'est pas une variable.

autorisé car `MAX_SIZE` n'est pas une variable mais sera remplacé par la constante `23`.

Sommaire chapitre 3

Tableaux multidimensionnels statiques

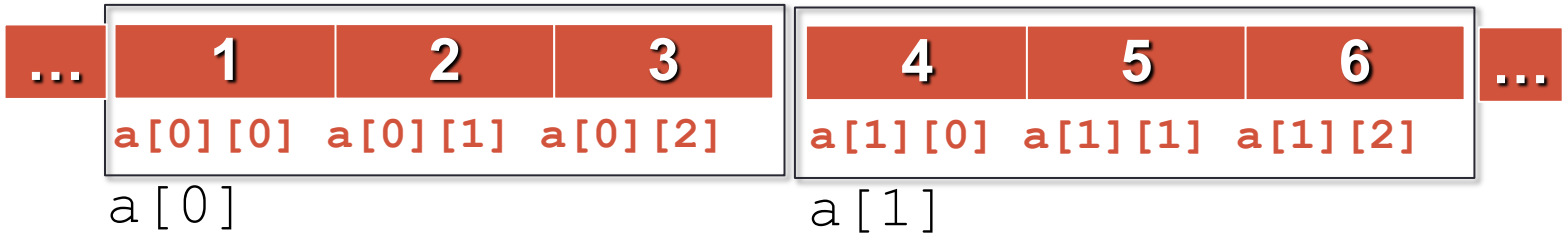
- Un tableau à 2 dimensions (L x C)

  est un tableau de L [tableaux de C éléments].

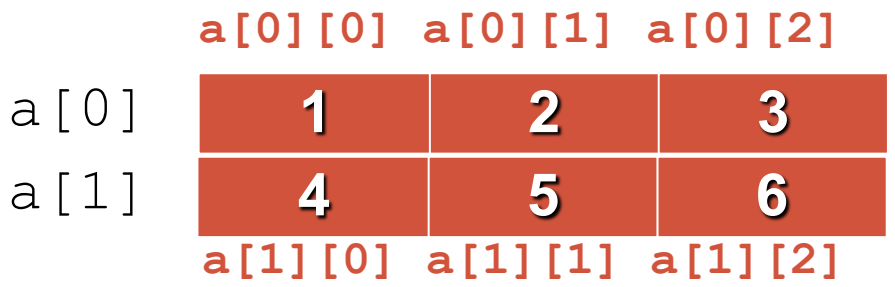
- Déclaration : `type name[L][C];` L lignes et C colonnes

  `int a[2][3]={{1,2,3},{4,5,6}};` avec initialisation

En mémoire:



- Représentation logique :



Tableaux multidimensionnels

- Accès aux composantes du tableau 2D :

```
a[i][j]=1;
```

avec $0 \leq i < L$

$0 \leq j < C$

met à 1 l'élément (`i`,`j`) du tableau

Tableaux multidimensionnels

```
int main(void)
{  int i,j;
   int lignes=2, colonnes=3;
   int m[2][3] = {{0,1,2},{3,4,5}};      /*m[lignes][colonnes]…*/
                                         /*…car variables*/
   m[1][0]= 10;

   for (i=0;i<2 /*ou lignes*/;i++)
   {
        for (j=0;j<3;j++)
        {  printf("%d ", m[i][j]);
        }
        printf("\n");
   }
   return 0;
}
```

Ce code affiche à l'écran :

```
 0   1   2
10   4   5
```

Tableaux multidimensionnels

```
#define NB_LINES 2
#define NB_COLS  3

int main(void)
{ int i,j;
  int m[NB_LINES][NB_COLS] = {{0,1,2},{3,4,5}};

  m[1][0]= 10;

  for (i=0;i<NB_LINES;i++)
  {
       for (j=0;j<NB_COLS;j++)
       {  printf("%d ", m[i][j]);
       }
       printf("\n");
  }
  return 0;
}
```

Définir des constantes : une bonne pratique

Ce code affiche à l'écran :

```
 0   1   2
10   4   5
```