

IN100 : Cours 8 – Les Tableaux

Sandrine Vial
`sandrine.vial@uvsq.fr`

Novembre 2015

Les Tableaux

- ▶ Ensemble d'éléments de **même type** désignés par une **unique variable** .
- ▶ Chaque élément est repéré par sa position dans le tableau

Un tableau d'entiers

12	34	45	5	234	6	-34	9	10
----	----	----	---	-----	---	-----	---	----

Déclaration

Pour définir un tableau il faut :

- ▶ un **nom** de variable,
- ▶ un **type** pour les éléments,
- ▶ une **taille fixe** : nombre (entier) d'éléments (de cases) dans le tableau.
- ▶ Par exemple :
 - ▶ Un tableau qui s'appelle T et qui contient 20 entiers :
`int T[20];`
 - ▶ Un tableau qui s'appelle TP et qui contient 30 points :
`POINT TP[30];`

Identification des éléments

- ▶ Un élément est accessible avec sa position dans le tableau.
- ▶ La première position dans le tableau porte le numéro **0**.
- ▶ Si on a un tableau de taille N , les éléments sont stockés entre les positions (indices) 0 et $N - 1$.

Un tableau de 9 entiers

	12	34	45	5	234	6	-34	9	10
Indices	0	1	2	3	4	5	6	7	8

Accès à un élément du tableau T déclaré par
`int T[10];`

- ▶ `T[0]` : premier élément du tableau T.
- ▶ `T[1]` : deuxième élément du tableau T.
- ▶ `T[9]` : dernier élément du tableau T.

Attention : `T[10]` n'existe pas !

Utilisation

- ▶ Une fois déclaré un tableau peut être vu comme une suite de variables indépendantes.

```
int T[10];
```

- ▶ Ici on a 10 variables entières :

$T[0], T[1], T[2], \dots, T[9]$

- ▶ Utilisation comme une variable habituelle :
 - ▶ $T[5] = 10 * 2 - 45;$
 - ▶ $T[4] = T[4] + 2;$
 - ▶ $T[2] = T[3] - 1;$

Expressions valides

```
int T[20];  
int n = 2;  
int p = 10;
```

```
T[3] = 5;  
T[5] = T[5] + 1;  
T[4] = T[4] - 6;  
T[n] = T[n] * 4;  
T[n+6] = 89;  
T[(n+p)%20] = 4;
```

Expressions valides

```
POINT TP[3];  
int n = 2;  
int p = 10;  
  
TP[0].x = 100;  
TP[0].y = 200;  
TP[1].x = 100;  
TP[1].y = 600;  
TP[2] = wait_cllic();
```


Initialisation des éléments d'un tableau

- Boucle pour initialiser chacune des valeurs

```
int main()
{
    int T[20];
    int i;

    for (i=0 ; i<20 ; i++)
    {
        T[i] = 0;
    }
}
```

Affichage des éléments d'un tableau

- Boucle pour afficher chacune des valeurs

```
int main()
{
    int T[20];
    int i;

    for (i=0 ; i<20 ; i++)
    {
        write_int(T[i]);
        write_text(" ");
    }
    writeln();
}
```

Tableaux et fonctions

- ▶ Paramètre formel :
`void f(int T[10])` ou `void f(int T[])`
- ▶ Paramètre effectif `f(T)` ;

Tableaux et fonctions

```
void Initialise_Tableau(int T[], int N)
{
    int i;

    for(i = 0; i < N; i++)
    {
        T[i] = 0;
    }
}

void Affiche_Tableau(int T[], int N)
{
    int i;

    for(i = 0; i < N; i++)
    {
        write_int(T[i]); write_text(" ");
    }
}

int main()
{
    int T[10];

    Initialise_Tableau(T,10);
    Affiche_Tableau(T,10);
}
```

Tableaux à 2 dimensions

- ▶ Déclaration : `int T[10][20];` : un tableau contenant 10×20 entiers.
- ▶ Accès à un élément `T[i][j]`
- ▶ Passage de paramètres `void f(int T[10][])`
- ▶ Appel de la fonction `f` : `f(T);`