

Tableaux unidimensionnels [tb]

Résumé de cours

Université de Haute Alsace

Unisciel 

algotprog 

Version 18 mai 2018

Table des matières

1	C - Résumé de cours	1
1.1	Définitions	1
1.2	Notations	2
1.3	Tableaux et paramètres	2
1.4	Parcours de tableaux	2

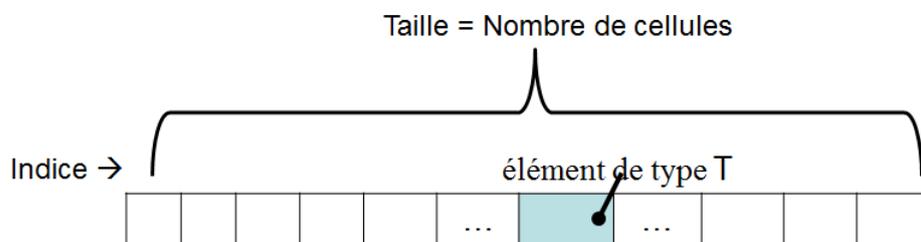
1 C - Résumé de cours

1.1 Définitions



Tableau

(Sous-entendu unidimensionnel ou linéaire) Collection **homogène indicée** sur \mathbb{N} , c.-à.d. une séquence d'éléments de **même type** portant tous le même nom et se distinguant les uns des autres par un **indice**.



Taille logique vs physique

Dite aussi taille **effective** (le nombre d'éléments effectivement utilisés) que l'on oppose à la taille **physique** (la taille maximale du tableau).

1.2 Notations

C/C++

Déclaration d'un tableau

```
TypeElement nomTab[taille];
```

C/C++

Déclaration et initialisation

```
TypeElement nomTab[taille] = {val1, ..., valN}; // taille explicite  
TypeElement nomTab[] = {val1, ..., valN}; // taille de la liste
```

C/C++

Accès indiciel

```
tab[ k ]
```



C/C++ : Débordement

Les éléments d'un tableau de taille `TMAX` sont indicés de 0 à `TMAX-1`.
Il n'y a pas de contrôle de débordement du tableau!

1.3 Tableaux et paramètres



Tableau et paramètres

```
enum {TMAX = ...};  
typedef T Tableau[TMAX]; // avec T un type quelconque  
  
... ssprg(T tab[TMAX], int n, ...) // tab en modification  
... ssprg(Tableau tab, int n, ...) // autre écriture  
  
... ssprg(const T tab[TMAX], int n, ...) // tab en lecture seule  
... ssprg(const Tableau tab, int n, ...) // autre écriture
```

1.4 Parcours de tableaux

La répétitive `Pour` est le moyen le plus simple de parcourir complètement un tableau. Le parcours partiel est le plus souvent basé sur une répétitive conditionnelle (`TantQue` ou `Répéter`). Les boucles imbriquées combinent généralement une répétitive `Pour` externe avec une répétitive conditionnelle interne.