Introduction à la programmation (C)

Cours 2 : les tableaux (statiques)

Tableaux (statiques)

Structure permettant de stocker plusieurs données du même type

* Accès facile à la $i$ème donnée (case)
* Taille fixe

Numéro de la case ou indice

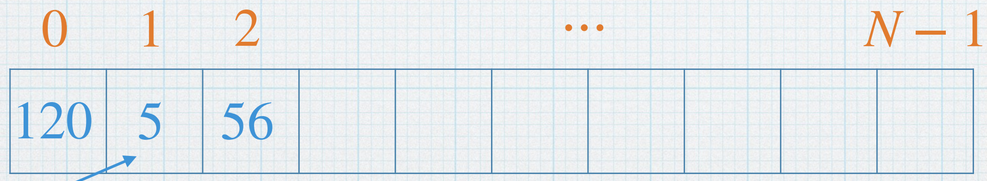

Tableau d'entiers

Valeur stockée
à la case d'indice 1

Les indices commencent à 0

Déclaration / accès

* De taille fixe (constante déclarée au début)
* Déclarés au même endroit que les variables (début de bloc)

Tableau nommé tab de N (constante) cases de type Type

```
Type tab [N] ;
```

Déclaration

```
tab[I]
```

Accès à la $i$ème case

Exemple

```
#include <stdio.h>
#include <math.h>
#define N 20

int main ()
{
    int i ;
    float T[N] ;
    for (i=0; i<N; i=i+1)
    {
        T[i] = sqrt(i);
    }
    return 0 ;
}
```

Création d'un tableau T de 20 réels contenant à la case i la donnée $\sqrt{i}$

A compiler avec l'option -lm

Déclaration/initialisation

Syntaxe pour la déclaration et initialisation simultanées :

Type tab [N] = { val1, …, valN } ;

```
#define N 5

int main ()
{
    int i ;
    char T[N] = {‘a’, ‘b’, ‘c’, ‘d’, ‘e’} ;
    return 0 ;
}
```

Tab = {v1, … vn} **impossible plus loin dans le code !**

Déclaration/initialisation

En C, impossible de récupérer la longueur d'un tableau à partir du tableau !

sizeof(type)

sizeof(int) → 4
sizeof(float) → 4
sizeof(double) → 8

~~sizeof(variable)~~

C++ seulement

Tableaux bi-dimensionnels

```
// Déclaration simple
Type tab [Nlignes] [Ncol] ;

// Déclaration + initialisation par liste des valeurs (par lignes)
Type tab [Nlignes] [Ncol]  = { val1, …., valM } ;  // M = Alignes x Ncol

// Déclaration + initialisation par ligne
Type tab [Nlignes] [Ncol]  = { {val1, …., valNcol} , {val1, …., valNcol } ;
```

```
#define Nl 2
#define Nc 3
int main ()
{
    char T[Nl][Nc] = {‘a’, ‘b’, ‘c’, ‘d’, ‘e’, ‘f’} ;
    return 0 ;
}
```

‘a’	‘b’	‘c’
‘d’	‘e’	‘f’