

UEO11 COURS/TD 1

Contenu du semestre

Cours et TDs sont intégrés

Le programme UEO12 est le suivant :

- rappel et complément sur les opérateurs et introduction aux tableaux 1D
- fonction, sous programme
- pointeurs, structure
- tri des tableaux
- fonctions récursives

S. REGIS
Chapitre 1

Rappel et complément sur les opérateurs et introduction aux tableaux

I Complément sur les opérateurs en C

Opérateurs binaires :

+ - * / % (reste de la division entière)

Opérateurs de comparaison :

== < <= > >= !=(différent)

Opérateurs logiques :

! négation d'une valeur logique

Ex : !(3>1) est donc la négation de vrai (1) c'est-à-dire faux (0)

&& Et de 2 valeurs logiques

|| Ou de 2 valeurs logiques

Opérateur conditionnel (peu utilisé mais à connaître)

(expression 1) ? (expression 2) : expression 3

Cet ensemble vaudra expression 2 si expression 1 \neq 0
expression 3 si expression 1 = 0

Ex : $((n > 0) ? n : -n)$ donne la valeur absolue de n (donner le détail svp)

Remarque :

Il existe d'autres types d'opérateurs comme les opérateurs binaires qui permettent de manipuler directement les bits mais ils ne seront pas vus dans ce cours.

Opérateur d'affectation

a=3 ;

b=a ;

c opérateur = expression est équivalent à écrire :

c = c opérateur expression

Ex :

s - = a ; \leftrightarrow s=s-a ;

p*=3 ; \leftrightarrow p=p*3 ;

incrémenter i++ (\leftrightarrow i=i+1)

décrémenter j-- (\leftrightarrow j=j-1)

Remarque

Il existe un ordre de priorité pour toutes ces opérateurs mais il ne sera pas présenté dans ce cours (travail perso).

II Introduction aux tableaux

1. Définitions – tableaux monodimensionnels (à 1 indice)

Les variables utilisées jusqu'à présent sont parfois inadaptées au traitement à réaliser
Il est difficile d'en utiliser un nombre important, particulièrement lorsqu'il s'agit de valeurs de même type.

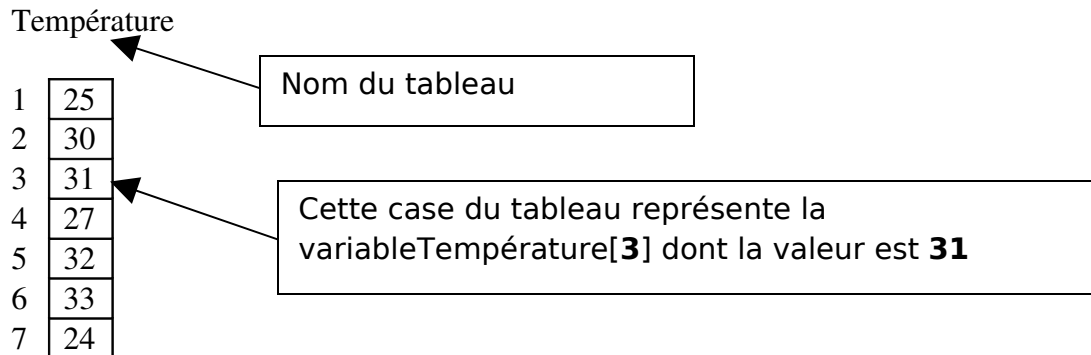
Exemples : relevés mensuels, statistiques journalières

Un tableau (encore appelé table ou variable indexée) est un ensemble de données, qui sont toutes de même type, désigné par un identificateur unique (le nom du tableau), et qui se distinguent les uns des autres par leur numéro d'indice.

Exemple : les températures sous abri à 15h00 des jours d'une semaine seront les 7 valeurs de la variable température, qui est un tableau de 7 éléments (variables) de type réel désigné par :

- Température[1], Température[2], ..., Température[7]

Représentation graphique :



L'utilisation d'un indice variable présente le principal intérêt des tableaux

Si la variable k entière a pour valeur 3, alors

- Température[k] = 23
- Température[k+1] = 36
- Température[k-1] = 30

Mais **attention** l'écriture de Température[-2] ou Température[12] n'ont pas de sens car elles font référence à des éléments inexistants

De même le calcul de la température moyenne de la semaine se fera de façon très simple (on suppose ici que le tableau est déjà rempli) :

Entrée

Sortie

Réel moyenne

Variables

Entiers somme, k

Début

somme \leftarrow 0

Pour k \leftarrow 1 **à** 7 **faire**

 somme \leftarrow somme + Température[k]

fpour

moyenne \leftarrow somme / 7

Ecrire (moyenne)

Fin

Comme tout objet, un tableau doit être déclaré avant tout utilisation

3 éléments fondamentaux définissent un tableau à un indice :

- Son nom : identificateur respectant les règles classiques des identificateurs d'un programme
- Le nombre de ses éléments (c'est la taille du tableau)
- Le type de données qu'il contient

Exemple :

Tableau de 7 entiers Température

Tableau de [1..7] d'entiers Température

Principe de déclaration d'un tableau 1D :

<type> <identificateur>[taille_i] ;

Ex : int A[10] ; //tableau A de 10 entiers

La taille correspond au nombre de cases du tableau

Attention : en langage C, les indices, permettant de localiser le contenu d'une case d'un tableau, varient entre

0 et taille-1

Remarque :

Lors de la déclaration d'un tableau, la taille d'un tableau doit être une valeur numérique et ne peut pas être une variable de type entier. La taille d'un tableau doit donc être constante (nous verrons dans un prochain chapitre qu'il est possible de contourner cette obligation en utilisant la notion de pointeur)

N'importe quelle référence à une case peut être utilisée comme une simple variable :

```
int A[10] ;  
float B[15] ;  
float y ;
```

```
y=1.5;  
A[0]=3;  
A[3]=A[0]+2;
```

```
B[10]=2.5;  
B[13]=2.5*2/y;
```

Revenons à l'exercice sur la moyenne des températures et écrivons le programme en C correspondant à l'algorithme précédent (en ajoutant cette fois les commandes pour remplir le tableau).

```
int main()
{
    int i, somme;
    float moyenne, temperature[7];
    for (i=0;i<7;i++)
    {
        printf("Temperature[%d]=",i);
        scanf("%d",&temperature[i]);
    }

    somme=0;
    for (i=0;i<7;i++)
        somme=somme+temperature[i];
    moyenne=somme/7;
    printf("la température moyenne de la semaine est %f\n",moyenne);
    return(0) ;
}
```

Ecrire un algorithme (facultatif) et un programme qui demande à l'utilisateur de rentrer les 7 températures d'une semaine dans un tableau de 7 réels ; puis qui donne la température maximal et la température minimale de la semaine en précisant le numéro du jour correspondant (0= lundi, 1=mardi,...,6=dimanche)

Reprenons l'exemple du tableau des températures. Supposons que l'on veuille cette fois enregistrer les températures toutes les heures pour chaque jour. Il faut donc maintenant un tableau à deux indices : un pour le jour (7 jours) et un pour l'heure (24 heures).

[illegible]

3																									
4																									
5																									
6																									
7																									

Plus généralement un tableau à deux dimensions i et j se présente sous la forme :

	1	2	...					j
1								
2								
.								
i								

En C, la déclaration de tableaux à plusieurs indices se fait de la même façon que pour un tableau à 1 indice.

Principe :

<type> <identificateur>[taille₁][taille₂]...[taille_k];

Exemple:

int a[13];

char b[8][5][10];

float d [6][15][9];

float x :

Là aussi n'importe quelle référence à une case peut être utilisée comme une simple variable :

d[2][9][6]= a[3]*2.1/7;

x= d[2][9][6]*3.14;

On rappelle que les indices, permettant de localiser le contenu d'une case d'un tableau, varient entre 0 et taille-1

Il est possible d'affecter un tableau (à 1 indice ou à plusieurs indices) à un ensemble de valeurs dès sa déclaration par :

<type><identificateur>[taille₁][taille₂]....={ val₁, val₂, ... };

Exemple :

int matrice[2][3]={ 1,2,3,4,5,6} \Leftrightarrow $\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$

Exercice :

Ecrire un programme permettant de rentrer les températures de chaque heure pour chaque jour de la semaine dans un tableau et qui calcule la moyenne des températures horaires pour chaque jour, et la valeur maximale parmi ces moyennes en précisant le numéro du jour (0=lundi, 1=mardi, etc.) de cette moyenne maximale.

Exercice

Pour décrire les polynômes à coefficients réels de la forme : $P(x) = a_0 + a_1 x + \dots + a_n x^n$, On va utiliser les tableaux vus dans le cours.

Ecrire un programme permettant à l'utilisateur d'introduire : le degré **n** d'un polynôme P (x), les n + 1 coefficients de P, une valeur réelle x. Le programme doit calculer et afficher, $y = P(x)$ par l'algorithme de Hörner. La taille du tableau contenant les coefficients est de 31 (le degré maximal d'un polynôme pour cet exercice est donc 30)

(Remarque : cet algorithme ne coûte que **n** multiplications, et **n** additions)

Algorithme de Hörner :

y <- a_n

POUR i DE (n- 1) A 0 FAIRE y <- y * x + a_i

Remarque :

Nous reviendrons un peu plus tard sur les tableaux, en voyant les méthodes de tri du tableau