

Cahier de TD n° 2

THEME 1 : Tableaux : définition de tableaux, taille maximum, taille utile, affichage, saisie.

Vrai / faux

Définition et syntaxe

saisie / affichage de tableaux

THEME 2 : tableaux : utilisation et algorithmes classiques

Recherche de la deuxième plus grande valeur d'un tableau

Suppression de doublons

Fusion de tableaux triés

THEME 3 : tableaux de caractères : travailler avec du texte

Message personnalisé

Noms de fichiers

THEME 4 : tableaux à plusieurs dimensions

Calcul matriciel

THEME 1 : Tableaux : définition de tableaux, taille maximum, taille utile, affichage, saisie.

Vrai / faux :

répondez aux questions suivantes par vrai ou faux

- Un tableau a toujours une taille maximum
- Un tableau a toujours une taille utile
- Un tableau se manipule comme une variable 'classique'
- Lors de la définition d'un tableau, on doit mettre une valeur numérique entre les crochets
- On peut utiliser une expression entière comme indice pour manipuler les variables contenues dans un tableau
- Les indices des variables d'un tableau de taille maximum N vont de 1 à N
- L'ordinateur vérifie systématiquement que les indices sont valides lors des manipulations des variables d'un tableau
- Un indice peut être un entier, un réel ou un caractère
- Un tableau ne peut contenir que des variables entières ou caractères
- On peut initialiser les variables d'un tableau lors de sa définition

Définition et syntaxe :

Parmi les définitions de tableaux suivantes, indiquez celles qui sont incorrectes et pourquoi. Si c'est possible, indiquez la définition qu'il aurait fallu écrire.

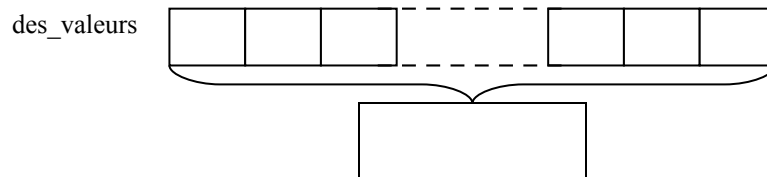
- `long tablo={1,2,3,4,5};`
- `double t[3];`
- `tableau x[124];`
- `double tab_reel[];`
- `long taille;`
`taille = 5;`
`char tabc[taille]={'a', 'b', 'c', 'd', 'e'};`
- `long vals[4]={1,2,3,4};`
- `long vals2[18000];`

saisie / affichage de tableaux

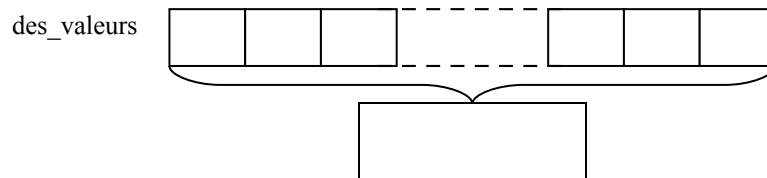
rappel : lorsque l'on indique que l'on saisit ou que l'on affiche un tableau, il faut comprendre : on saisit des valeurs à ranger dans les variables d'un tableau, on affiche les valeurs des variables contenues dans un tableau. De plus, on utilise quasiment systématiquement des boucles lorsque l'on traite un tableau.

Que fait le programme suivant ? Remplissez les schémas au fur et à mesure, et complétez les instructions sur fond gris à la fin du programme.

```
Int main()  
{  
double des_valeurs[10];  
long taille_max, taille;  
long compt;
```



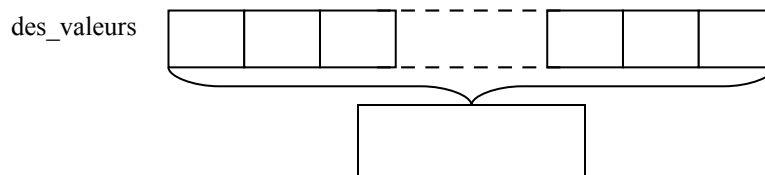
```
taille_max = 10;  
des_valeurs[0] = 0.05;  
taille = 1;
```



taille_max

taille

```
for (compt =1 ; compt <= taille_max-1; compt++)  
{  
    des_valeurs[compt] = des_valeurs[compt-1] + 0.1;  
}
```



```
// affichage de toutes les valeurs du tableau
```

```
for   
{  
    printf("la variable %ld du tableau vaut :%ld\n",  
    );  
}
```

- Ecrire un programme qui fait la saisie de valeurs de type caractère et qui les range dans un tableau contenant au maximum 50 caractères. L'utilisateur peut arrêter la saisie en saisissant le caractère '&'. Avant de commencer le programme, écrivez bien quelles sont les conditions auxquelles on arrête (ou on continue) la boucle dans laquelle se fait la saisie.
- Ecrire un programme qui fait la saisie de valeurs de type réel, qui les range dans un tableau et qui effectue les calculs suivants :
 - a) s'il y a plus de 3 valeurs : calcul de la somme des valeurs absolues
 - b) s'il y a exactement 2 valeurs : calcul de leur moyenne harmonique \bar{h} :

$$\frac{2}{\bar{h}} = \frac{1}{x_1} + \frac{1}{x_2}$$

où x_1 et x_2 sont les deux valeurs du tableau.

- c) s'il y a exactement 4 valeurs : calcul de leur moyenne géométrique \bar{g} :

$$\bar{g} = \sqrt[4]{x_1 \cdot x_2 \cdot x_3 \cdot x_4}$$

où x_1, \dots, x_4 sont les quatre valeurs du tableau.

Pour le calcul de la racine quatrième : calculer la racine quatrième revient à élever le nombre à la puissance $\frac{1}{4}$, soit 0,25. On utilisera une instruction de calcul : *puissance(x,y)* qui calcule le nombre x (réel) élevé à la puissance y (nombre réel), et qui donne un résultat réel.

THEME 2 : tableaux : utilisation et algorithmes classiques

Recherche de la deuxième plus grande valeur d'un tableau

On considère un tableau (peu importe le type des variables qu'il contient, c'est à vous de le choisir) comportant au maximum N variables et dont M variables sont utilisées. En vous inspirant de l'algorithme de recherche de la valeur maximum, écrivez un programme qui recherche la deuxième plus grande valeur du tableau (celle qui se rapproche le plus du maximum sans l'atteindre).

Suppression de doublons

Soit un tableau contenant des variables entières. Ecrivez un programme à qui l'on fournit une valeur entière et qui supprime tous les exemplaires sauf un de cette valeur si elle se trouve en plusieurs exemplaires dans le tableau.

A partir du programme précédent, écrire un programme qui supprime tous les doublons (exemplaires multiples d'une valeur) d'un tableau.

Fusion de tableaux triés

On dispose de deux tableaux triés, on désire réunir les valeurs de ces deux tableaux dans un troisième tableau qui devra lui aussi être trié. On veut éviter de recopier les valeurs du premier tableau, puis à la suite les valeurs du deuxième tableau puis de trier le tout. Proposez un algorithme qui insère les valeurs dans la troisième tableau directement à leur bonne place, et écrivez le programme correspondant.

THEME 3 : tableaux de caractères : travailler avec du texte

Message personnalisé

Ecrire un programme qui saisit un nom d'utilisateur (sous la forme d'un texte) et qui répond par un message de bienvenue comportant le nom de l'utilisateur saisi précédemment. Le nom de l'utilisateur doit se trouver au milieu du message de bienvenue.

Noms de fichiers

Lorsque l'on écrit des programmes utilisant des fichiers, on accède à ces fichiers en précisant leur nom sous la forme de texte, donc de tableaux de caractères. En général, l'utilisateur entre un nom de fichier court (ou relatif par rapport au répertoire courant), mais l'ordinateur a besoin de connaître le nom long ou absolu du fichier.

Ecrire un programme qui saisit un nom de disque dur (sous la forme d'une lettre pour désigner le disque), le nom d'un répertoire où sont stockés des fichiers de l'utilisateur (ce peut être un nom comportant plusieurs répertoires si on a besoin de parcourir plusieurs niveaux dans l'arborescence), et le nom d'un fichier (celui que veut manipuler l'utilisateur). Le programme doit afficher le nom absolu (nom long) du fichier.

THEME 4 : tableaux à plusieurs dimensions

Calcul matriciel

Une matrice en deux dimensions est définie en mathématiques comme un ensemble de coefficients repérés par leur numéro de ligne et numéro de colonne. Une matrice M à n lignes et p colonnes est un ensemble de coefficients m_{ij} , avec i compris entre 1 et n et j compris entre 1 et p .

$$M = \underbrace{\left(\begin{array}{ccc} m_{11} & m_{12} & m_{1p} \\ m_{21} & \dots & \\ & & \\ m_{n1} & & m_{np} \end{array} \right)}_{p \text{ colonnes}} \left. \vphantom{\begin{array}{ccc} m_{11} & m_{12} & m_{1p} \\ m_{21} & \dots & \\ & & \\ m_{n1} & & m_{np} \end{array}} \right\} n \text{ lignes}$$

On peut naturellement les représenter par des tableaux à 2 dimensions.

Si M est une matrice ayant le même nombre de lignes que de colonnes (c'est à dire que $n=p$), on dit que la matrice est carrée et on peut calculer sa trace $\sum_{i=1}^n m_{ii}$

a) Ecrire un programme qui initialise une matrice carrée avec des valeurs aléatoires et qui calcule sa trace.

On peut multiplier entre elles deux matrices A et B pour obtenir le produit $A.B$ (attention avec les matrices la multiplication n'est plus une opération commutative) à la seule condition que le nombre de colonnes de la matrice A soit égal au nombre de lignes de la matrice B .

Si A est une matrice à n lignes et p colonnes et B une matrice à p lignes et q colonnes, alors le produit $A.B$ est une troisième matrice (nommons la C) à n lignes et q colonnes

$$\forall i \in [1..n], \forall j \in [1..q], c_{ij} = \sum_{k=1}^p a_{ik} \cdot b_{kj}$$

Où a_{ik} , b_{kj} et c_{ij} sont des coefficients des matrices A , B et C .

b) Ecrivez un programme qui initialise deux matrices avec des entiers au hasard et qui réalise le produit de ces deux matrices. Vous pouvez vous aider d'un schéma pour matérialiser comment se fait le produit avant de tenter d'écrire le programme.