Ministère de l'enseignement supérieur et de la recherche scientifique
Université Abderrahmane MIRA de Bejaia

Faculté de Technologie
Département de Technologie

Structure des ordinateurs et applications

À propos du cours

- **Crédits**: 02
- **Coefficients**:02

Les tableaux

❑ ***Introduction***

Jusqu'à présent, on a étudié les variables simples (entier, réel, caractère, chaîne, booléen).Chaque variable simple correspond à un seul espace mémoire et ne peut contenir qu'une seule valeur à un moment donné.

Mais lorsque l'on veut représenter plusieurs valeurs du même type (comme une liste de notes, un ensemble de nombres, etc.), les variables simples ne suffisent plus. Dans ce cas, on utilise des **variables indicées**, appelées aussi **tableaux.**

Définition du tableau:

Un tableau est une structure de données homogène qui permet de stocker plusieurs valeurs du même type sous un nom unique. Il existe deux types de tableaux :

- les tableaux a une dimension (Vecteurs)
- les tableaux a deux dimensions (Matrices).

Les trois éléments qui définissent un tableau sont :

- ✓ Le nom du tableau;
- ✓ Le type des données qu'il contient ;
- ✓ La taille du tableau (nombre de cases).

Tableau à une dimension (Vecteur)

Un tableau à une dimension, ou vecteur, est un ensemble d'éléments de même type, accessibles grâce à un seul indice qui indique leur position.

Syntaxe (Déclaration)

Algorithmique	En langage C
constante MAX = <valeur_entière> variable <id_tab>:Tableau[0..MAX-1] de <type>;	const int MAX = <valeur_entière>; <type> <id_tab>[MAX];

Exemple

constante
MAX = 50
variable
V:Tableau[0..49] de réel;

const int MAX = 50;
real V[MAX];

En langage C, la **taille maximale** d'un vecteur ainsi que les variables ***n*** (taille) et ***i*** (indice) doivent toujours être déclarées de ***type int.***

En langage C, la taille d’un vecteur ainsi que l’indice doivent être ***des valeurs entières***. L’indice d’un tableau commence à **0** et se termine à **n-1**, où **n** représente la taille du tableau. Cela permet de parcourir correctement tous les éléments du vecteur.

En algorithme, on écrit explicitement les indices (0..MAX-1).En C, on ne précise que la taille, les indices sont définis automatiquement par le langage (de 0 à MAX-1).

Ce n’est pas obligatoire d'utiliser une constante pour la taille maximale de tableau ;

La plage d'indice *0* .. *MAX-1* représente l'indice du premier élément *0* et l'indice du dernier élément;

C'est pas obligatoire d'utiliser toutes les composantes du tableaux, pour cela on déclare une variable entière N qui représente la taille du tableau à utilise ;

Directement lors de la déclaration → {val1, val2, …}Partiellement → les cases restantes = 0; Case par case après déclaration → v[i] = valeur.

Accès à un élément

En programmation un élément de tableau est désigné par le nom du tableau suivi de l'indice (la position) de l'élément dans le tableau (c'est la même syntaxe que nous utilisons en langage C). Soit :

<Nom du Tableau>[<Indice de l'élément>]

Exemple : notes[0];

Initialisation d'un tableau unidimensionnel

Un tableau peut être initialisé en utilisant deux méthodes :

- En utilisant des affectations ;
- En utilisant la lecture à partir d'un fichier de données ou à partir de clavier.

Par affectations : L'affectation permet de modifier la valeur d'un élément du vecteur en utilisant son indice.

Syntaxe

id_tab[indice] = valeur ;

id_tab[indice1] = id_tab[indice2];

id_tab[indice]= Expression ;

id_tab[indice]=Constante ;

Forme	***Signification***
id_tab[i] = id_tab[j];	Copie d'une case vers une autre
id_tab[i] = a + b;	Affectation d'un calcul
id_tab[i] = 10; id_tab[indice] = valeur	Affectation d'une constante Affectation d une valeur

Par lecture : On utilise pour cela l'opération de lecture ***scanf***: c'est la méthode la plus pratique.

Lecture d’un élément/Lecture d’un vecteur

La lecture d’un élément d’un vecteur consiste à lire une composante du vecteur en utilisant son indice. La lecture d’un vecteur, quant à elle, consiste à lire l’ensemble de ses éléments et ce, à l’aide d’une boucle. Comme le nombre d’éléments du vecteur est connu, la boucle pour est la plus adaptée pour effectuer cette opération.

Algorithmique

```
Lire (n) ;
Pour i←1 à N faire
    Lire (vecteur[i]);
FinPour;
```

En langage C

```
Scanf(‘’%d”,&n)
for(i = 0; i < =N; i++) {
    scanf("%d", &vecteur[i]);
}
```

Écriture d'un vecteur

L'écriture d'un vecteur permet d'afficher l'ensemble de ses éléments. Elle s'effectue également à l'aide d'une boucle.

Algorithmique	En langage C

Pour i←1 à N faire
  Écrire (vecteur[i]);
FinPour ;
	for(i = 0; i < N; i++) {
	  printf("%d ", vecteur[i]);}

Lecture et écriture d'un tableau de *N* élément

```
Algorithme : LireEcrireTableau
Variables
V : Tableau[0..99] de réel
N, i : entier
Début
Écrire("Introduire nombre d'éléments N :");
Lire(N);
Écrire("Introduire V :");
Pour i ← 0 à N-1 faire
        Lire(V[i]);
FinPour;
Écrire("Affichage de V :");
Pour i ← 0 à N-1 faire
        Écrire(V[i]);
FinPour;
Fin
```

```
#include <stdio.h>
int main()
{
float V[100];
int N, i;
printf("Introduire nbre
d'éléments N : \n");
scanf ("%d" , &N);
printf("Introduire V : \n");
for (i=0; i<N ; i++)
scanf ("%f", &V[i]);
printf("Affichage du V : \n") ;
for (i=0; i<N ; i++)
printf ("%.2f", V[i]);
}
```

❑ *Manipulation d'un vecteur*

Somme de deux Vecteurs V1 et V2

pour i←0 **à** N-1 faire
V[i] ← V1[i] + V2[i];
FinPour;

```
for (i=0; i<N ; i++)
V[i] = V1[i] + V2[i];
```

Produit scalaire de deux vecteurs V1 et V2

PS ← 0;
pour ←0 **à** N-1 **faire**
PS ← PS + V1[i] * V2[i];
FinPour;

```
Ps = 0;
for (i=0; i<N ; i++)
ps += V1[i]*V2[i];
```

❑ *Manipulation d'un vecteur*

Recherche d'un élément dans un tableau

```
i ← 1 ;
Trouve ← Faux;
TantQue (i<=N) et (Trouve=Faux) faire
Si V[i] = Val Alors
Trouve ← Vrai;
R ← i;
FinSi
i ← i + 1;
Fin-Tant-Que;
```

```
i=0; trouve=false;
while ((i<n) &&
(!trouve)){
if (V[i] == val){
trouve=true; pos = i;
}
i++;
}
```

Recherche du maximum dans un tableau unidimensionnel

```
max ← Notes[0]; imax ← 0;
pour i←1 à N-1 faire
si Notes[i] > max alors
max ← Notes[i];
imax ← i;
fin-si;
Fin-Pour;
```

```
max=Notes[0] ; imax=0;
for (i=1; i<n; i++)
{
if (Notes[i] > max) {
max = Notes[i]; imax=i;
}
}
```