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

Tableau à une dimension (Vecteur)

Un tableau à deux dimensions, ou matrice, est un ensemble d'éléments de même type, repérés par deux indices.
Ces indices permettent d'indiquer la position de chaque élément dans le tableau : le premier indice correspond généralement à la ligne et le second à la colonne. Ils permettent ainsi d'accéder directement à la valeur située à l'intersection d'une ligne et d'une colonne données..

Syntaxe (Déclaration)

Algorithmique	**En langage C**
Variables id_tab : Tableau [0..maxLigne-1, 1.. maxCol-1] de type ;	Type id_tab[maxLigne][maxCol];

Exemple

Variables
Variables Notes : Tableau [1..30, 1.. 20] d entier;

int notes[30][20];

Accès à un élément

L’accès à un élément d’une matrice se fait en utilisant deux indices : le premier pour la ligne et le second pour la colonne, selon la syntaxe suivante :

id_tab[indL][indC].

Exemple : int notes[3][4];

Affectation d’un élément

L’affectation permet de modifier la valeur d’un élément de la matrice en utilisant ses deux indices.

Syntaxe

id_tab[indL, indC] ← valeur ;
id_tab[indL, indC] ← id_tab2[indL, indC]
id_tab[indL, indC] ← Expression ;
id_tab[indL, indC] ← Constante ;

Écriture générale	Exemple avec A[3][3]	Explication
A[i][j] = valeur	A[1][2] = 5;	On met une valeur directe dans une case
A[i][j] = B[i][j]	A[1][2] = B[1][2];	On copie la valeur d’un autre tableau
A[i][j] = expression	A[i][j] = i + j;	On met le résultat d’un calcul
A[i][j] = constante	A[i][j] = 0;	On met la même valeur dans les cases

Lecture d’un élément/ Lecture de la matrice

La lecture d’un élément d’une matrice consiste à lire une composante située à l’intersection d’une ligne et d’une colonne, en utilisant ses deux indices.

Pour lire ou afficher les éléments d’un tableau, on utilise généralement une boucle "**pour**" (pour, for, etc.). Cette structure permet de parcourir chaque élément du tableau de manière ordonnée, en utilisant un indice.

Algorithmique

```
Lire (N, M);
pour i←0 à N-1 faire
Pour j←0 à M-1 faire
        Lire (mat[i, j]);
FinPour;
FinPour;
```

En langage C

```
scanf("%d %d", &n, &m);
for (i=0; i<n; i++)
  for (j=0; j<m; j++)
    scanf("%f", &mat[i][j]);
   }
}
```

Écriture d’un vecteur

L’écriture d’une matrice permet d’afficher l’ensemble de ses éléments.
Elle s’effectue également à l’aide de deux boucles imbriquées, afin de parcourir successivement les lignes et les colonnes de la matrice.

Algorithmique

```
pour i←0 à N-1 faire
Pour j←0 à M-1 faire
        Écrire (mat[i, j]);
FinPour;
FinPour;
```

En langage C

```
for (i=0; i<n; i++)
   for (j=0; j<m; j++)
    printf("%8.2f", mat[i][j]);
  }
}
```

Programme langage c pour la lecture et affichage d'une matrice

Lecture et affichage d'une matrice

```
algorithme LectureAffichageMatrice
Variables
mat:tableau [0..99, 0..99] de réel;
N,M, i,j:entier;
Début
Lire (N, M);
pour i←0 à N-1 faire
    Pour j←0 à M-1 faire
        Lire (mat[i, j]);
    FinPour;
FinPour;
pour i←0 à N-1 faire
    Pour j←0 à M-1 faire
        Écrire (mat[i, j]);
    FinPour;
FinPour;
Fin
```

```
#include <stdio.h>
int main() {
    float mat[100][100];
    int i, j, n, m;
    scanf("%d %d", &n, &m);
for (i = 0; i < n; i++)
        for (j = 0; j < m; j++)
            scanf("%f", &mat[i][j]);
for (i = 0; i < n; i++) {
        for (j = 0; j < m; j++)
            printf("%8.2f", mat[i][j]);
        printf("\n");
    }

    return 0;
}
```

Programme langage c pour la Somme de deux matrices

❑ *Manipulation d'un vecteur*

Somme de deux matrices

```
pour i←0 à N-1 faire
Pour j←0 à M-1 faire
C[i,j] ← A[i,j] + B[i,j];
FinPour;
FinPour;
```

```
for(i = 0; i < n; i++) {
for(j = 0; j < m; j++) {
C[i][j] = A[i][j] + B[i][j];
}
}
```

❑ *Manipulation d'un vecteur*

Produit d'une matrice par un vecteur

pour i←0 **à** N-1 **faire**
p[i] = 0;
Pour j←0 **à** M-1 **faire**
P[i] ← p[i] + mat[i, j]*vect[j];
FinPour;
FinPour;

```
for (i=0; i<n; i++) {
p[i] = 0;
for (j=0; j<m; j++)
p[i] += mat[i][j] *
vect[j];
}
```

Programme langage c pour compter le nombre d'éléments négatifs, positifs et nuls dans une matrice

❑ *Manipulation d'un vecteur*

Compter le nombre d'éléments négatifs, positifs et nuls dans une matrice

```
nbP ← 0;nbN ← 0;nbNul ← 0;

pour i ← 0 à N-1 faire
    pour j ← 0 à M-1 faire
        Si mat[i, j] > 0 Alors
            nbP ← nbP + 1;
        Sinon
            Si mat[i, j] < 0 Alors
                nbN ← nbN + 1
            Sinon
                nbNul ← nbNul + 1;
            FinSi;
        FinSi;
    FinPour;
FinPour;
```

```
fnbP = 0;nbN = 0;nbNul = 0;

for (i = 0; i < n; i++) {
    for (j = 0; j < m; j++) {
        if (mat[i][j] > 0) {
            nbP++;
        } else if (mat[i][j] < 0) {
            nbN++;
        } else {
            nbNul++;
        }
    }
}
```

Traduire l'algorithme en programme C

```
Algorithme Exercice_01;
Variables  T : Tableau [1..100, 1..100] d'entier;
                N, i, j, P : entier;

Début
    // Entrés
 Ecrire('Entrez la taille (N x N) de la matrice T: ');
 Lire (N);
 Ecrire('Entrez les éléments de la matrice T : ');
  Pour  i ← 1 à N faire
      Pour  j ← 1 à N faire
          Lire(T[i, j]);
      FinPour ;
  FinPour ;
    //Traitement
   P ← 1;
    Pour  i ← 1 à N faire
        P←P* T[i, i];
    FinPour ;
    // Sortie
   Ecrire ('P= ', P);
Fin.
```

Exercice 1:

Soit A une matrice de dimension N×M et de type réel. Ecrire un programme en C permettant de trouver et d'afficher la composante maximale de la matrice A ainsi que sa position (ligne et colonne).

Exercice 2:

Soit une matrice A réelle d'ordre N × M.

1. Ecrire un algorithme/programme en C qui calcule la somme et la moyenne des éléments de la matrice A.
2. Ecrire un algorithme/programme en c qui permet de calculer la somme de chaque ligne et le produit de chaque colonne.

Exercice 3

Soit A et B deux matrices carrées d'ordre N et M.

Ecrire un algorithme/programme en C qui permet de calculer le produit de A et B.