

Les tableaux à deux dimensions (Matrices)

Université A-Mira de Bejaia-2020/2021

Département de Technologie

1ere année Technologie

Module : Informatique2

Présenté par : Mme MAMMERI

Soit la structure A suivante :

Indices

i

j

	1	2	3	4
1	2	5	9	-1
2	3	-7	8	12
3	0	6	-4	1

- Cette structure est un tableau à deux dimensions (matrice) composée de trois lignes et de quatre colonnes. Sa taille est 3×4 .
- Cette matrice est donc composée de 3×4 éléments (12 éléments)
- Pour accéder à un élément donné dans une matrice, on doit indiquer le numéro de la ligne et le numéro de la colonne de cet élément, comme exemple :
 - On accède au 1^{er} élément de la matrice A par $A[1,1]$. $A[1,1]=2$
 - On accède au dernier élément de A par $A[3,4]$. $A[3,4]=1$
- Pour parcourir les éléments d'une matrice, on utilise deux indices i et j , i pour les lignes et j pour les colonnes.

Déclaration d'une matrice

En algorithmique

Variable id_variable :tableau[1..taille_max_ligne,1..taille_max_colonne]de type ;

En programmation Pascal

Var id_variable :array[1..taille_max_ligne,1..taille_max_colonne]of type ;

➤ *Exemple :*

En algo : variable A:tableau[1..10,1..10]de reel;

En pascal : var A: array[1..10,1..10]of real;

Lecture des éléments d'une matrice

En algorithmique :

```
Pour i ← 1 à nb_ligne faire  
  Pour j ← 1 à nb_colonne faire  
    Lire([A[i,j]) ;  
  Finpour ;  
Finpour ;
```

nb_ligne est le nombre de lignes de la matrice A à remplir.
nb_ligne ≤ taille_max_ligne déclaré

nb_colonne est le nombre de colonnes de la matrice A à lire.
nb_colonne ≤ taille_max_colonne déclaré

En Pascal:

```
for i := 1 to nb_ligne do  
  for j := 1 to nb_colonne do  
    read([A[i,j]) ;
```

Affichage des éléments d'une matrice

En algorithmique :

```
Pour i ← 1 à nb_ligne faire
  Pour j ← 1 à nb_colonne faire
    écrire([A[i,j]]);
  Finpour;
Finpour ;
```

En Pascal:

```
for i := 1 to nb_ligne do
begin
  for j := 1 to nb_colonne do
    write([A[i,j], ' ');
  Writeln;
End;
```

Algorithme de lecture et d'affichage d'une matrice d'ordre $n \times m$

Algorithme lecture_affichage;

Variables A : Tableau [1..10, 1..10] de Réel

i, j, n, m : entier

Debut

 Lire(n, m)

 Pour $i \leftarrow 1$ à n faire

 Pour $j \leftarrow 1$ à m faire

 Lire(A[i, j])

 Fin-Pour

 Fin-Pour

 Pour $i \leftarrow 1$ à n faire

 Pour $j \leftarrow 1$ à m faire

 Ecrire(A[i, j])

 Fin-Pour

 Fin-Pour

Fin .

Déroulement pour n=2 et m=4

Instructions	n	m	i	j	A	Affichage								
Lire(n, m)	2	4												
Pour i=1 Pour j=1 Lire(A[1,1])			1	1	<table border="1"><tr><td>2</td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr></table>	2								
2														
Pour j=2 Lire (A[1,2])				2	<table border="1"><tr><td>2</td><td>3</td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr></table>	2	3							
2	3													
Pour j=3 Lire (A[1,3])				3	<table border="1"><tr><td>2</td><td>3</td><td>-1</td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr></table>	2	3	-1						
2	3	-1												
Pour j=4 Lire (A[1,4])				4	<table border="1"><tr><td>2</td><td>3</td><td>-1</td><td>5</td></tr><tr><td></td><td></td><td></td><td></td></tr></table>	2	3	-1	5					
2	3	-1	5											
Pour i=2 Pour j=1 Lire(A[2,1])			2	1	<table border="1"><tr><td>2</td><td>3</td><td>-1</td><td>5</td></tr><tr><td>4</td><td></td><td></td><td></td></tr></table>	2	3	-1	5	4				
2	3	-1	5											
4														
Pour j=2 Lire (A[2,2])				2	<table border="1"><tr><td>2</td><td>3</td><td>-1</td><td>5</td></tr><tr><td>4</td><td>6</td><td></td><td></td></tr></table>	2	3	-1	5	4	6			
2	3	-1	5											
4	6													
Pour j=3 Lire (A[2,3])				3	<table border="1"><tr><td>2</td><td>3</td><td>-1</td><td>5</td></tr><tr><td>4</td><td>6</td><td>0</td><td></td></tr></table>	2	3	-1	5	4	6	0		
2	3	-1	5											
4	6	0												
Pour j=4 Lire (A[2,4])				4	<table border="1"><tr><td>2</td><td>3</td><td>-1</td><td>5</td></tr><tr><td>4</td><td>6</td><td>0</td><td>8</td></tr></table>	2	3	-1	5	4	6	0	8	
2	3	-1	5											
4	6	0	8											
Pour i=1 Pour j=1 Ecrire (A[1, 1]) Pour j=2 Ecrire (A[1, 2]) Pour j=3 Ecrire (A[1, 3]) Pour j=4 Ecrire (A[1, 4]) Pour i=2 Pour j=1 Ecrire (A[2, 1]) Pour j=2 Ecrire (A[2, 2]) Pour j=3 Ecrire (A[2, 3]) Pour j=4 Ecrire (A[2, 4])			1	1		2 3 -1 5 4 6 0 8								

Quelques Algorithmes/programmes sur les matrices

- La somme des éléments d'une matrice A de type réels :

Soit A=

2	5	1
3	-2	8
0	6	-4

La somme des éléments de la matrice A est :

$$S=2+5+1+3-2+8+0+6-4$$

$$S=A[1,1]+A[1,2]+A[1,3]+ A[2,1]+A[2,2]+A[2,3]+ A[3,1]+A[3,2]+A[3,3]$$

- Dans le cas d'une matrice de taille nxm , n est le nombre de lignes et m est le nombre de colonnes introduits par l'utilisateur

$$S= A[1,1]+A[1,2]+A[1,3]+ \dots+A[1,m]+A[2,1]+A[2,2]+A[2,3]+ \dots+A[2,m]+ \dots+A[n,1]+A[n,2]+A[n,3]+ \dots+A[n,m]$$

$$S = \sum_{i=1}^n \sum_{j=1}^m A[i,j]$$

Algorithme somme ;**Variables** *A*:tableau [1..10,1..10] de reel;*i*, *j*, *n*, *m* : entier ;*S*: reel;**Debut**Lire(*n*,*m*) ;*pour i*←1 à *n faire**pour j*←1 à *m faire**lire (A [i,j]) ;**finpour ;**finpour ;**s*←0;*pour i*←1 à *n faire**pour j*←1 à *m faire**s*←*s*+*A*[*i*,*j*] ;*finpour ;**finpour ;*Ecrire('la somme =', *s*) ;**Fin.****program somme ;****Uses** wincrt;**Var***A*: array[1..10,1..10] of real;*i*, *j*, *n*, *m* : integer ;*S*: real;**Begin**

Write ('introduire le nombre de lignes et de colonnes : ');

read(*n*,*m*) ;

Write ('introduire les éléments de la matrice A: ');

for *i*:=1 to *n* dofor *j*:=1 to *m* do*Read*(*A* [*i*,*j*]) ;*s*:=0;for *i*:=1 to *n* dofor *j*:=1 to *m* do*s*:=*s*+*A*[*i*,*j*] ;Write ('la somme =', *s*) ;**end.**

➤ Somme de deux matrices

➤ Écrire un algorithme/programme PASCAL qui permet de réaliser la somme de deux matrices réelles A et B d'ordre nxm.

Pour réaliser la somme de deux matrices, les deux matrices doivent être de même type et de même dimension, la matrice résultat C est calculée comme suit :

A=

2	5	1
3	-2	8
0	6	-4

B=

1	-1	4
0	2	1
0	3	5

C=

3	4	5
3	0	9
0	9	1

$$C[1,1]=A[1,1]+B[1,1]$$

$$C[1,2]=A[1,2]+B[1,2]$$

$$C[1,3]=A[1,3]+B[1,3]$$

.

.

.

$$C[3,3]=A[3,3]+B[3,3]$$

$$\text{Donc : } C[i,j]=A[i,j]+B[i,j], \quad i=1..n, j=1..m$$

```

Algorithme Somme_Matrice;
Variables A,B,C : Tableau [1..10, 1..10] de Réel;
i, j,n, m : entier;
Début
Lire(n, m) ;
Pour i←1 à n faire
    Pour j←1 à m faire
        Lire( A[i, j] ) ;
    Fin-Pour ;
Fin-Pour ;
Pour i←1 à n faire
    Pour j←1 à m faire
        Lire(B[i, j]) ;
    Fin-Pour;
Fin-Pour ;
Pour i←1 à n faire
    Pour j←1 m faire
        C[i,j] ← A[i,j] + B[i,j] ;
    Fin-Pour ;
Fin-Pour;
Pour i←1 à n faire
    Pour j←1 à m faire
        Écrire( C[i, j] )
    Fin-Pour ;
Fin-pour;
Fin .

```

```

Program Somme_Matrice;
Uses wincrt ;
var A,B,C : array[1..10,1..10] of Real;
i,j, n, m : integer ;

Begin
Write('Donner la taille de A et B : ');
    Read(n, m);
Writeln('Donner les valeurs de A : ');
For i:=1 to n do
    For j:=1 to m do
        Read(A[i,j]);
Writeln('Donner les valeurs de B : ');
For i:=1 to n do
    For j:=1 to m do
        Read(B[i,j]);
For i:=1 to n do
    For j:=1 to m do
        C[i,j] := A[i,j]+B[i,j];
Writeln('La matrice C = A+B :');
For i:=1 to n do
Begin
    For j:=1 to m do
        write(C[i,j]:8:2);
        Writeln;
end;
End.

```

➤ Transposé d'une matrice

Soit la matrice A de taille $N \times M$ de type réel

Le transposé de la matrice A d'ordre $N \times M$ est une matrice A^t d'ordre $M \times N$. Chaque ligne de A devient une colonne de A^t (ou chaque colonne de A devient une ligne pour A^t).

➤ Exemple

$A =$

2	5	-2	1
3	-2	7	8
0	6	9	-4

$A^t =$

2	3	0
5	-2	6
-2	7	9
1	8	-4

Dans cet exemple :

A est d'ordre (3×4)

A^t est d'ordre (4×3)

$A[3,2]=6$

Cet élément se trouve à la position $(2,3)$ dans la matrice transposée A^t

$A^t[2,3]=6$

▪ chaque élément dans A de position (i,j) se trouve à la position (j,i) dans A^t

Algorithme Transposé

Variables A, At : Tableau [1..10, 1..10] de Réel

i, j, n, m : entier

Début

Lire(n, m)

Pour $i \leftarrow 1$ à n faire

 Pour $j \leftarrow 1$ à m faire

 Lire(A[i, j])

 Fin-Pour

Fin-Pour

Pour $i \leftarrow 1$ à m faire

 Pour $j \leftarrow 1$ à n faire

At[i, j] \leftarrow A[j, i]

 Fin-Pour

Fin-Pour

Pour $i \leftarrow 1$ à m faire

 Pour $j \leftarrow 1$ à n faire

 ecrire(At[i, j])

 Fin-Pour

Fin-Pour

Fin.