

TP Informatique 2 – Semestre 2

Série de TP N°2 – Tableaux à deux dimensions - Matrices

Exercice N°01 : Algorithme → Programme

Soit l'algorithme suivant :

Algorithme Matrice;

Variables

i,j,n,m : Entier;

A : Tableau[1..100, 1..100] d'entier;

MIN : Tableau[1..100] d'entier;

Début

{-*-*- Entrées -*-*-}

Écrire('Donner le nombre des lignes et des colonnes');

Lire(n,m);

Écrire('Donner les composantes de la matrice A');

Pour i←1 **à** n **faire**

Pour j←1 **à** m **faire**

Lire (A[i,j]);

FinPour;

FinPour;

{-*-*- Traitements -*-*-}

Pour j←1 **à** m **faire**

MIN[j] ← A[1, j];

Pour i←2 **à** n **faire**

Si (A[i, j] < MIN[j]) **alors**

MIN[j] ← A[i, j];

FinSi;

FinPour;

FinPour;

{-*-*- sorties -*-*-}

Écrire('Les composantes de la matrice A sont: ');

Pour i←1 **à** n **faire**

Pour j←1 **à** m **faire**

Écrire(A[i,j] , ' ');

FinPour;

FinPour;

Écrire('Les composantes de vecteur Min sont : ');

Pour j←1 **à** m **faire**

Écrire(MIN[j], ' ');

FinPour;

Fin.

Questions :

1- Traduire l'algorithme en Programme PASCAL.

2- Compiler et exécuter le programme pour :

N = 2, M=4 et

A=

5	40	-10	-6
1	8	0	-8

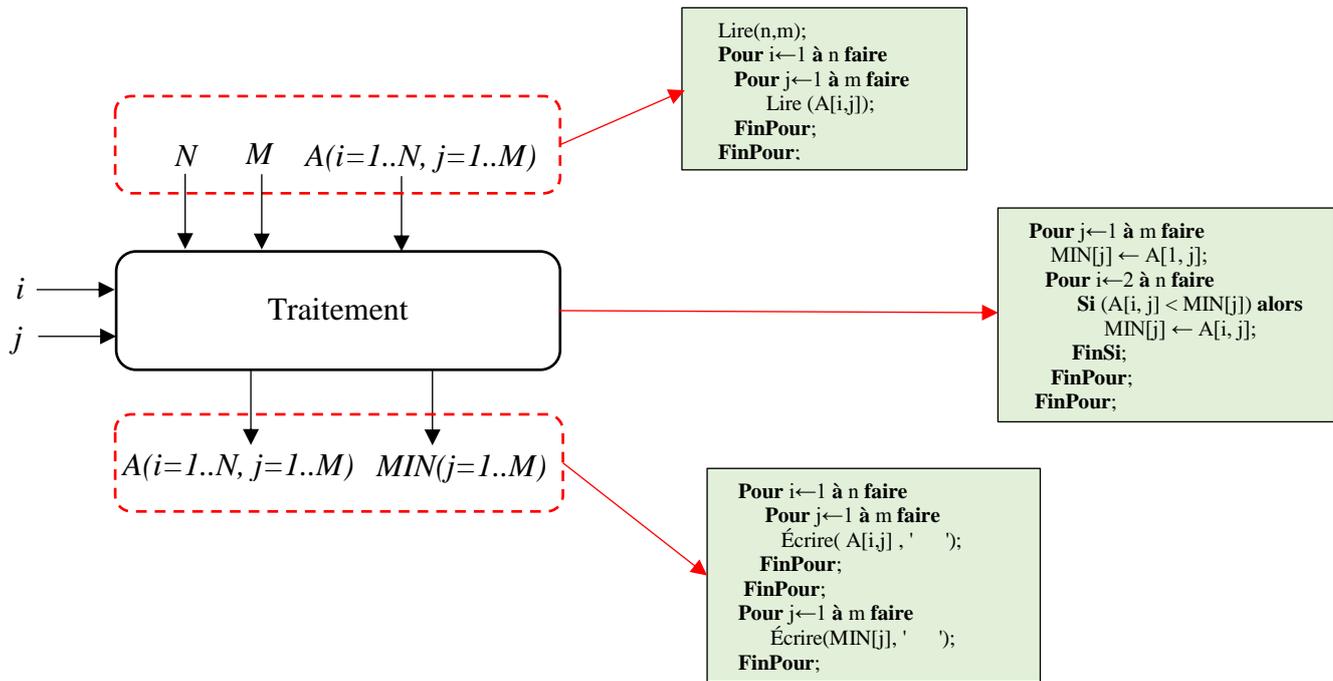
3- Dérouler le programme pour les valeurs de N, M et A ci-dessus ?

4- Déduire ce que fait le programme ?

5- Ré-écrire le programme en remplaçant la boucle *Pour* par la boucle *Tantque* dans la partie **traitements** et par la boucle *Répéter* dans la partie **sorties**.

Solution :

Le schéma des variables d'entrées, variables de sorties et la partie traitement est le suivant :



Remarque :

Les variables i et j sont des variables de traitement ou intermédiaires, utilisées pour parcourir la matrice A.

1 - Algorithme/programme PASCAL :

Algorithme	Programme PASCAL
<p>Algorithme Matrice;</p> <p>Variabes i, j, n, m : Entier; A : Tableau[1..100, 1..100] d'entier; MIN : Tableau[1..100] d'entier;</p> <p>Début</p> <p>{Les entrées} Écrire('Donner le nombre des lignes et des colonnes de A'); Lire(n,m); Écrire('Donner les composantes de la matrice A');</p> <p>Pour $i \leftarrow 1$ à n faire Pour $j \leftarrow 1$ à m faire Lire (A[i,j]); FinPour; FinPour;</p> <p>{Les traitements} Pour $j \leftarrow 1$ à m faire MIN[j] ← A[1, j]; Pour $i \leftarrow 2$ à n faire Si (A[i, j] < MIN[j]) alors MIN[j] ← A[i, j]; FinSi; FinPour; FinPour;</p> <p>{Les sorties} Écrire('Les composantes de la matrice A sont: ');</p>	<p>program Matrice;</p> <p>var i, j, n, m : integer; A : array[1..100, 1..100] of integer; MIN : array[1..100] of integer;</p> <p>begin</p> <p>{Les entrées} WriteLn('Donner le nombre des lignes et des colonnes de A'); ReadLn (n,m); WriteLn('Donner les composantes de la matrice A');</p> <p>For $i:=1$ to n do {La boucle des lignes} For $j:=1$ to m do {La boucle des colonnes} read (A[i,j]);</p> <p>{Les traitements} For $j:=1$ to m do Begin MIN[j] := A[1, j]; For $i:=2$ to n do If(A[i, j]<MIN[j])then MIN[j] := A[i,j]; End;</p> <p>{Les sorties} WriteLn('Les composantes de la matrice A sont: '); For $i:=1$ to n do Begin</p>

Pour i←1 à n faire Pour j←1 à m faire Écrire(A[i,j], ' '); FinPour ; FinPour ; Écrire('Les composantes de vecteur Min sont : '); Pour j←1 à m faire Écrire(MIN[j], ' '); FinPour ; Fin .	For j:=1 to m do Write (A[i,j], ' '); {Afficher dans la même ligne} WriteLn; {Sauter la ligne} End ; WriteLn('Les composantes de vecteur Min sont: '); For j:=1 to m do Write (MIN[j], ' '); {Afficher dans la même ligne} End .
---	---

2 - Compiler et exécuter le programme pour : N = 2, M=4 et

A=

5	40	-10	-6
1	8	0	-8

```

1 Program Matrice;
2 Var
3   i,j,N,M : integer;
4   A : array [1..100,1..100] Of integer;
5   MIN : array [1..100] Of integer;
6 Begin
7   {Les entrées}
8   writeln('Donner le nombre des lignes et des colonnes : ');
9   readln(N,M);
10  writeln('Donner les composantes de la matrice A : ');
11  For i:=1 To N Do
12    For j:=1 To M Do
13      read(A[i,j]);
14
15  {Les traitements}
16  For j:=1 To M Do
17    Begin
18      MIN[j] := A[1,j];
19      For i:=2 To n Do
20        If (A[i,j]<MIN[j]) Then
21          MIN[j] := A[i,j];
22    End;
23
24  {Les sorties}
25  writeln('Les composantes de la matrice A sont : ');
26  For i:=1 To N Do
27    Begin
28      For j:=1 To M Do
29        write(A[i,j], ' ');
30      writeln();
31  writeln('Les composantes de vecteur MIN sont : ');
32  For j:=1 To M Do
33    write(MIN[j], ' ');
34 End.

```

MyPascal V1.20.5 (Exécution) C:\Users\Ahm...
Donner le nombre des lignes et des colonnes :
2 4
Donner les composantes de la matrice A :
5 40 -10 -6
1 8 0 -8
Les composantes de la matrice A sont :
5 40 -10 -6
1 8 0 -8
Les composantes de vecteur MIN sont :
1 8 -10 -8

Après l'exécution

3 - Dérouler le programme pour les valeurs de N, M et A ci-dessus ?

Instructions	Variables				A	MIN	Affichage															
	n	m	i	j																		
Readln (n,m);	2	4	/		/	/																
For i:=1 to n do For j:=1 to m do read (A[i,j]) ;	"	"	1 2 3 4	1 2 3 4	<table border="1" style="display: inline-table; margin-left: 20px;"> <tr> <td></td> <td>j=1</td> <td>j=2</td> <td>j=3</td> <td>j=4</td> </tr> <tr> <td>i=1</td> <td>5</td> <td>40</td> <td>-10</td> <td>-6</td> </tr> <tr> <td>i=2</td> <td>1</td> <td>8</td> <td>0</td> <td>-8</td> </tr> </table>		j=1	j=2	j=3	j=4	i=1	5	40	-10	-6	i=2	1	8	0	-8	/	
	j=1	j=2	j=3	j=4																		
i=1	5	40	-10	-6																		
i=2	1	8	0	-8																		

<pre> For j:=1 Begin MIN[j] := A[1, j]; For i:=2 to n do If(A[i, j]<MIN[j])then I<5 True MIN[j] := A[i,j] ; MIN[1] := 1; End; </pre>	"	"	2	1	"	<table border="1"> <tr> <td style="background-color: green;">█</td> <td></td> <td></td> <td></td> </tr> </table>	█							
█														
<pre> For j:=2 Begin MIN[j] := A[1, j]; For i:=2 to n do If(A[i, j]<MIN[j])then 8<40 True MIN[j] := A[i,j] ; MIN[2] := 8; End; </pre>	"	"	2	2	"	<table border="1"> <tr> <td>1</td> <td style="background-color: green;">█</td> <td></td> <td></td> </tr> </table>	1	█						
1	█													
<pre> For j:=3 Begin MIN[j] := A[1, j]; For i:=2 to n do If(A[i, j]<MIN[j])then 0<-10 False MIN[j] := A[i,j] ; MIN[3] := -10; End; </pre>	"	"	2	3	"	<table border="1"> <tr> <td>1</td> <td>8</td> <td style="background-color: green;">█</td> <td></td> </tr> </table>	1	8	█					
1	8	█												
<pre> For j:=4 Begin MIN[j] := A[1, j]; For i:=2 to n do If(A[i, j]<MIN[j])then -8<-6 True MIN[j] := A[i,j] ; MIN[4] := -8; End; </pre>	"	"	2	4	"	<table border="1"> <tr> <td>1</td> <td>8</td> <td>-10</td> <td style="background-color: green;">█</td> </tr> </table>	1	8	-10	█				
1	8	-10	█											
<pre> WriteLn('Les composantes de la matrice A sont:'); For i:=1 to n do Begin For j:=1 to m do Write (A[i,j], ' '); WriteLn(); End; </pre>	"	"	1 2	1 2 3 4	"	<p>Les composantes de la matrice A sont:</p> <table border="1"> <tr> <td>5</td> <td>40</td> <td>-10</td> <td>-6</td> </tr> <tr> <td>1</td> <td>8</td> <td>0</td> <td>-8</td> </tr> </table>	5	40	-10	-6	1	8	0	-8
5	40	-10	-6											
1	8	0	-8											
<pre> WriteLn('Les composantes de vecteur Min sont:'); For j:=1 to m do Write (MIN[j], ' '); End. </pre>	"	"		1 2 3 4	"	<p>Les composantes de vecteur Min sont:</p> <table border="1"> <tr> <td>1</td> <td>8</td> <td>-10</td> <td>-8</td> </tr> </table>	1	8	-10	-8				
1	8	-10	-8											

4 - Dédurre ce que fait le programme ?

Le programme permet de chercher le minimum de chaque colonne de la matrice A et le sauvegarde dans un vecteur MIN de taille m.

5 - Ré-écrire le programme en remplaçant la boucle Pour par la boucle Tantque dans la partie traitements et par la boucle Répéter dans la partie sorties.

Programme PASCAL

```
program Matrice;
var
i,j,n,m :integer;
A : array[1..100, 1..100] of integer;
MIN : array[1..100] of integer;
begin

  {Les entrées}
  WriteLn('Donner le nombre des lignes et des colonnes de A');
  Readln (n,m);
  WriteLn('Donner les composantes de la matrice A');
  For i:=1 to n do
    For j:=1 to m do
      read (A[i,j] ); {Read pour lire dans la même ligne}

  {Les traitements}
  j:=1;
  while (j<=m) do
  Begin
    MIN[j] := A[1, j];
    i:=2;
    while (i<=n) do
    Begin
      If(A[i, j]<MIN[j])then
        MIN[j] := A[i,j] ;
      i:=i+1;
    End;
    j:=j+1;
  End;

  {Les sorties}
  WriteLn('Les composantes de la matrice A sont: ');
  i:=1 ;
  repeat
    j:=1;
    repeat
      Write ( A[i,j] , ' '); {Afficher dans la même ligne}
      j:=j+1 ;
    until (j>m) ;
    WriteLn() ; {Sauter la ligne}
    i:=i+1 ;
  until (i>n) ;
  WriteLn('Les composantes de vecteur Min sont: ');
  j :=1 ;
  repeat
    Write (MIN[j] , ' '); {Afficher dans la même ligne}
    j:=j+1 ;
  until (j>m) ;

End.
```

Exercice N°02 :

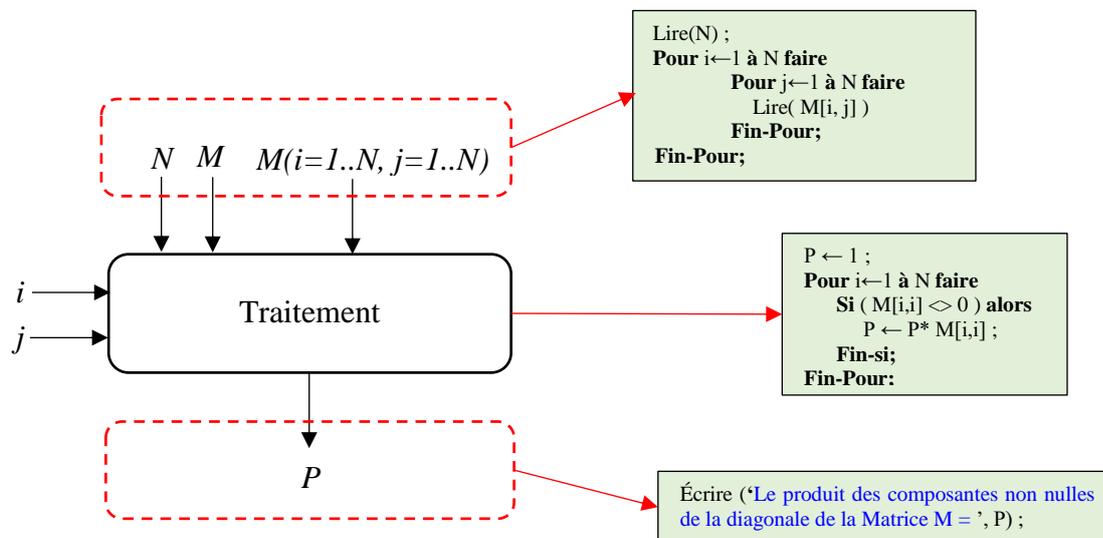
Soit M une matrice carrée de taille N x N et de type réel.

- Écrire un programme pascal qui permet de calculer le produit des composantes non nulles de la diagonale principale de la matrice M.
- Vérifier si la matrice M est symétrique.

Rappel : Une matrice M est symétrique si $M[i, j] = M[j, i]$ pour tout i et j .

Solution :

1.
Le schéma des variables d'entrées, variables de sorties et la partie traitement est le suivant :



Algorithme/programme PASCAL :

Algorithme	Programme PASCAL
<p>Algorithme exo2_1;</p> <p>Variabes M : Tableau [1..10,1..10] de réel; N,i,j : entier ;</p> <p>Début {Les entrées} Lire(N); Pour i←1 à N faire Pour j←1 à N faire Lire(M[i, j]) Fin-Pour; Fin-Pour; {Les traitements} P ← 1; Pour i←1 à N faire Si (M[i,i] <> 0) alors P ← P* M[i,i]; Fin-si; Fin-Pour;</p>	<p>Program exo2_1;</p> <p>Var M : Array [1..10,1..10] of real; N,i,j : integer ; P: real;</p> <p>Begin {Les entrées} Writeln('Donner la dimension de la matrice carrée:'); Read(N); Writeln('Donner les composantes de la matrice A : '); For i :=1 to N do For j :=1 to N do read(M[i, j])</p> <p>{Les traitements} P:=1; For i :=1 to N do Begin If (M[i,i] <> 0) then P := P* M[i,i];</p>

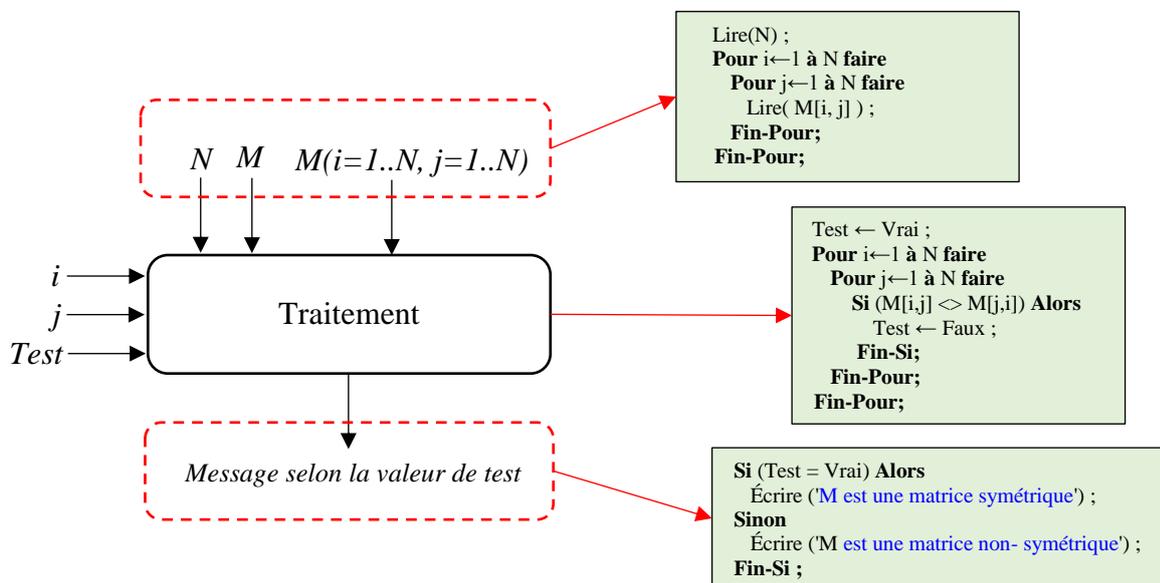
<p>{Les sorties}</p> <p>Écrire ('Le produit des composantes non nulles de la diagonale de la Matrice M = ', P) ;</p> <p>Fin.</p>	<p>End;</p> <p>{Les sorties}</p> <p>Writeln ('Le produit des composantes non nulles de la diagonale de la Matrice M = ', P) ;</p> <p>End.</p>
--	--

2. Rappel : T est symétrique si $M[i, j] = M[j, i]$ pour tout i et j .

Les étapes à suivre :

- D'abord supposer que M est symétrique (Test = True)
- Ensuite, comparer chaque case M [j, i] avec la case M [i, j].
- Si elles sont différentes alors affecter la valeur False à la variable Test.
- A la fin, il suffit de voir la valeur de Test pour savoir si la matrice M est symétrique ou pas.

Le schéma des variables d'entrées, variables de sorties et la partie traitement est le suivant :



Algorithme/programme PASCAL :

Algorithme	Programme PASCAL
<p>Algorithme exo2_2;</p> <p>Variabes M : Tableau [1..10,1..10] de réel; N,i,j : entier ; Test : booléen ;</p> <p>Début</p> <p>{Les entrées} Lire(N) ; Pour i←1 à N faire Pour j←1 à N faire Lire(M[i, j]) ; Fin-Pour ; Fin-Pour ;</p> <p>{Les traitements}</p>	<p>Program exo2_2;</p> <p>Var M : array [1..10,1..10] of real; N,i,j : integer ; Test : boolean ;</p> <p>Begin</p> <p>{Les entrées} Writeln('Donner la dimension de la matrice carrée:'); Read(N) ; Writeln('Donner les composantes de la matrice A : '); For i :=1 to N do For j :=1 to N do read(M[i, j]);</p> <p>{Les traitements}</p>

<pre> Test ← Vrai ; Pour i←1 à N faire Pour j←1 à N faire Si (M[i,j] <> M[j,i]) Alors Test ← Faux ; Fin-Si; Fin-Pour; Fin-Pour; {Les sorties} Si (Test = Vrai) Alors Écrire ('M est une matrice symétrique') ; Sinon Écrire ('M est une matrice non-symétrique') ; Fin-Si ; Fin. </pre>	<pre> Test := True ; For i:=1 to N do Begin For j:=1 to N do If (M[i,j] <> M[j,i]) Then Test := False ; End; End; {Les sorties} If (Test = True) then Writeln ('M est une matrice symétrique') Else Writeln ('M est une matrice non-symétrique') ; End. </pre>
---	---

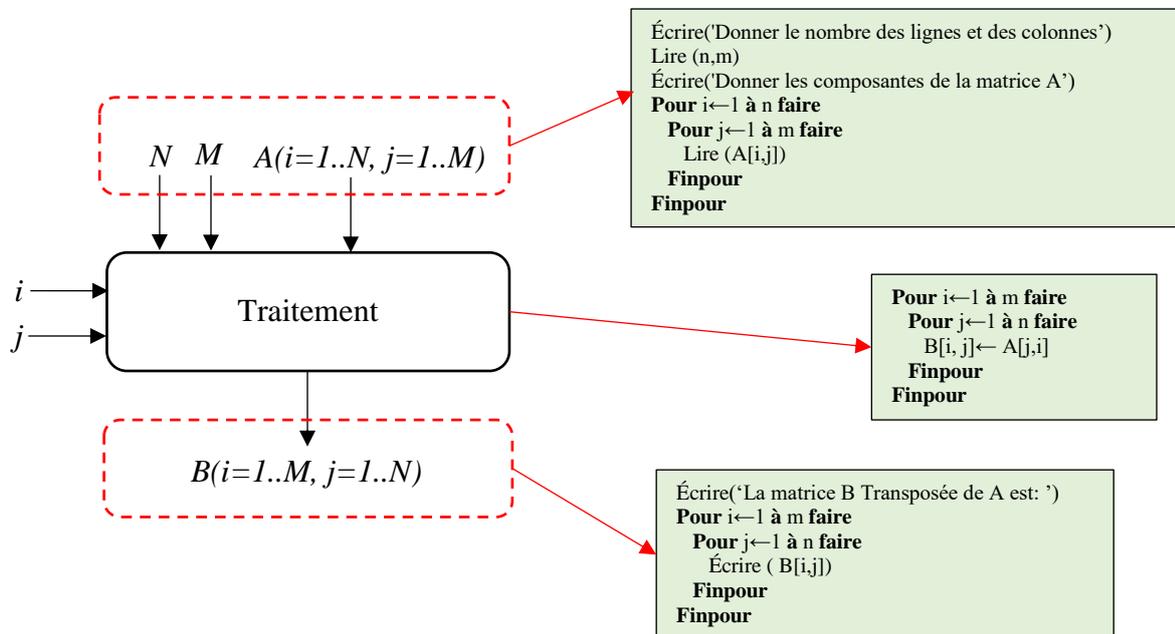
Exercice N°03 : Transposée d'une matrice

Écrire un algorithme/programme PASCAL qui permet de calculer la matrice B transposée d'une matrice réelle A d'ordre N x M.

Solution :

Le transposé d'une matrice A d'ordre N x M est un matrice B d'ordre M x N.

Chaque ligne de A devient une colonne de B (ou chaque colonne de A devient une ligne pour B). Chaque case B[i, j] correspond à la case A[j, i] tel que : i=1, ..., M et j=1, ..., N



Algorithme/programme PASCAL :

Algorithme	Programme PASCAL
Algorithme Transposee Variables i,j,n,m : Entier A, B : tableau[1..100, 1..100] de reel Debut	program Transposee; var i,j,n,m : integer; A,B : array[1..100, 1..100] of real; begin

<p>{Les entrées} Écrire('Donner le nombre des lignes et des colonnes de A') Lire (n,m) Écrire('Donner les composantes de la matrice A') Pour i←1 à n faire Pour j←1 à m faire Lire (A[i,j]) Finpour Finpour</p> <p>{Les traitements} Pour i←1 à m faire Pour j←1 à n faire B[i, j]← A[j,i] Finpour Finpour</p> <p>{Les sorties} Écrire('La matrice B Transposée de A est: ') Pour i←1 à m faire Pour j←1 à n faire Écrire (B[i,j]) Finpour Finpour Fin.</p>	<p>{Les entrées} WriteLn('Donner le nombre des lignes et des colonnes de A'); Readln (n,m); WriteLn('Donner les composantes de la matrice A'); For i:=1 to n do For j:=1 to m do read (A[i,j]) ; {Read pour lire dans la même ligne} End End</p> <p>{Les traitements} For i:=1 to m do Begin For j:=1 to n do B[i, j] := A[j,i] ; End; End;</p> <p>{Les sorties} WriteLn('La matrice B Transposée de A est:'); For i:=1 to m do Begin For j:=1 to n do write (B[i,j]:6:2) ; {Afficher dans la même ligne} writeLn; {Sauter la ligne} End; End; End.</p>
--	---

Exercice : Le Min et le Max dans une matrice et leurs positions

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

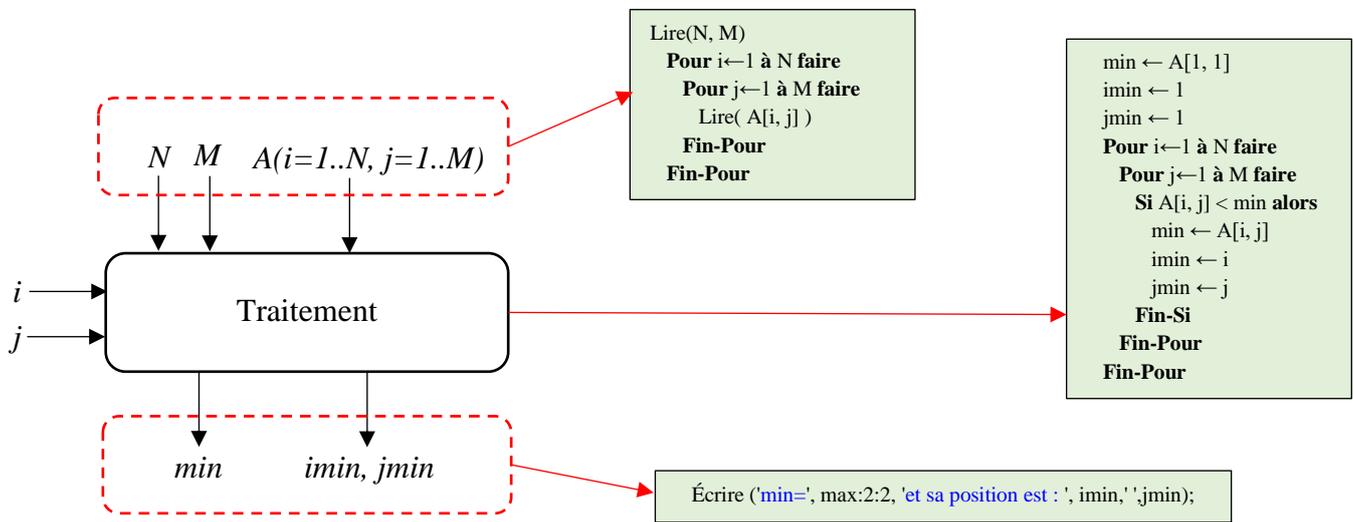
1. Écrire un algorithme/programme PASCAL qui permet de rechercher le plus petit élément dans la matrice A ainsi que sa position.
2. Écrire un algorithme/programme PASCAL qui permet de rechercher le plus grand élément dans la matrice A ainsi que sa position.

Solution :

1. Pour répondre à la première question, nous suivons la logique suivante :

On suppose que la première composante (A[1,1]) de la matrice A est le minimum, sa position est (imin=1, jmin=1), puis on parcourt les autres composantes de la matrice A et de les comparer à la première composante, dès qu'un élément est inférieur au minimum, ce dernier sera mis à jour ainsi que sa position imin (ligne) et jmin (colonne).

Le schéma des variables d'entrées, variables de sorties et la partie traitement est le suivant :



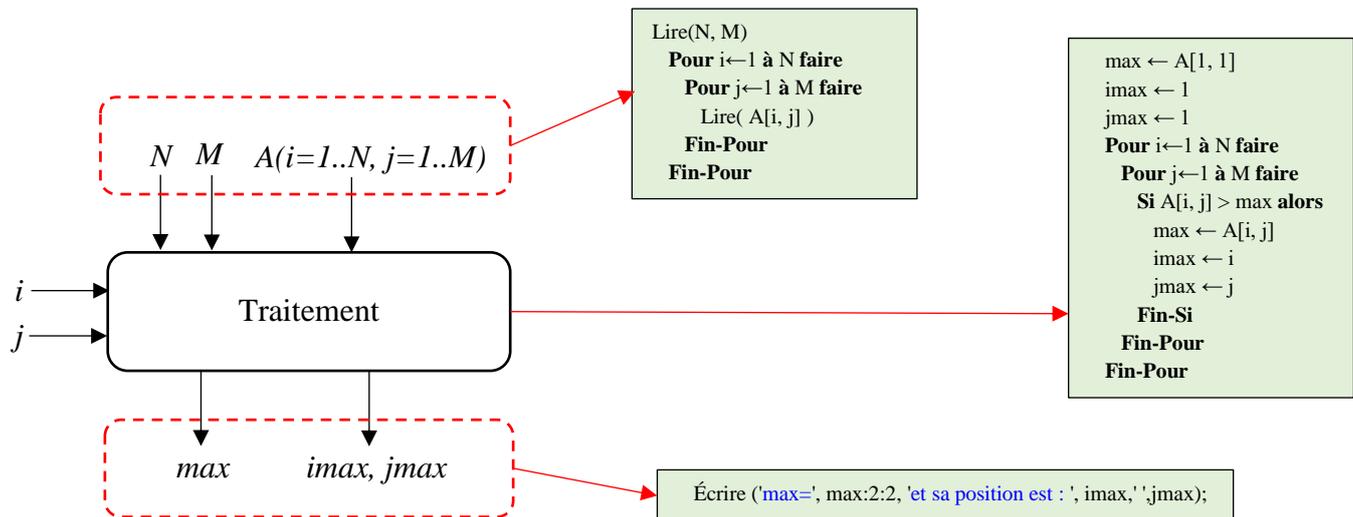
Algorithme/programme PASCAL :

Algorithme	Programme PASCAL
<p>Algorithme exo2_1</p> <p>Variables A : Tableau [1..10, 1..10] de Réel i, j, N, M, imin, jmin : entier Min : Réel</p> <p>Début Lire(N, M) Pour i←1 à N faire Pour j←1 à M faire Lire(A[i, j]) Fin-Pour Fin-Pour min ← A[1, 1] imin ← 1 jmin ← 1 Pour i←1 à N faire Pour j←1 à M faire Si A[i, j] < min alors min ← A[i, j] imin ← i jmin ← j Fin-Si Fin-Pour Fin-Pour Écrire('min=', min:3:2, 'et sa position est : ', imin, ' ', jmin); Fin</p>	<pre> Program exo2_1; Uses wincrt ; var A : array[1..10, 1..10] of Real; i, j, N, M, imin, jmin : integer; min : real; Begin {Les entrées} Write('Donner les dimensions de la Matrice A : '); Read(N, M); Writeln('Donner les composantes de la matrice A : '); For i:=1 to N do For j:=1 to M do Read(A[i, j]); {Les traitements} min := A[1, 1]; {On suppose que la première case est le min} imin:= 1; {Donc sa position est la ligne 1} jmin := 1; {et la colonne 1} For i:=1 to N do {Pour toute ligne i} For j:=1 to M do {Pour toute colonne j} if A[i, j] < min Then begin min := A[i,j]; {On actualise le Min} imin := 1; {On actualise sa ligne imin} jmin := 1; {On actualise sa colonne jmin} end; {Les sorties} Write('min=', min:3:2, 'et sa position est : ', imin, ' ', jmin); End. </pre>

2. Pour répondre à la deuxième question, nous suivons la logique suivante :

On suppose que la première composante (A[1,1]) de la matrice A est le maximum, sa position est (imax=1, jmax=1), puis on parcourt les autres composantes de la matrice et de les comparer à la première composante, dès qu'un élément est supérieur au maximum, ce dernier sera mis à jour ainsi que sa position imax (ligne) et jmax (colonne).

Le schéma des variables d'entrées, variables de sorties et la partie traitement est le suivant :



Algorithme/programme PASCAL :

Algorithme	Programme PASCAL
<p>Algorithme exo2_2</p> <p>Variables A : Tableau [1..10, 1..10] de Réel i, j, N, M, imax, jmax : entier max: Réel</p> <p>Début Lire(N, M) Pour i ← 1 à N faire Pour j ← 1 à M faire Lire(A[i, j]) Fin-Pour Fin-Pour max ← A[1, 1] imax ← 1 jmax ← 1 Pour i ← 1 à N faire Pour j ← 1 à M faire Si A[i, j] > max alors max ← A[i, j] imax ← i jmax ← j Fin-Si Fin-Pour Fin-Pour Écrire ('max=', max:2:2, 'et sa position est : ', imax, ', ', jmax); Fin</p>	<p>Program exo2_2; Uses winCRT ; var A : array[1..10, 1..10] of Real; i, j, N, M, imax, jmax : integer; max : real; Begin {Les entrées} Write('Donner les dimensions de la Matrice A : '); Read(N, M); Writeln('Donner les composantes de la matrice A : '); For i:=1 to N do For j:=1 to M do Read(A[i, j]); {Les traitements} Max := A[1, 1]; {On suppose que la première case est le max} imax:= 1; {Donc sa position est la ligne 1} jmax := 1; {et la colonne 1} For i:=1 to N do {Pour toute ligne i} For j:=1 to M do {Pour toute colonne j} if A[i, j] > max Then begin max := A[i,j]; {On actualise le Max} imax := 1; {On actualise sa ligne imax} jmax := 1; {On actualise sa colonne jmax} end; {Les sorties} Write('max=', max:2:2, 'et sa position est : ', imax, ', ', jmax); End.</p>