

Partie B :

Exercice N°01 : Algorithme → Programme PASCAL

Soit l'algorithme suivant :

Algorithme Matrice ;

Variables

A : Tableau [1..100, 1..100] de réel;

i, j, N : entier;

S, M : réel;

Début

{*-*- Entrées *-*-}

Ecrire('Donner la taille de la matrice carrée A :');

Lire(N);

Ecrire('Donner les composantes de la matrice A :');

Pour i ← 1 à N faire

Pour j ← 1 à N faire

 Lire(A[i,j]);

FinPour;

FinPour;

{*-*- Traitement *-*-}

S ← 0;

Pour i ← 1 à N faire

 S ← S+A[i,i] ;

FinPour;

M ← S/N ;

{*-*- Sorties *-*-}

Ecrire('S=', S:4:2, 'M=', M:4:2);

Fin.

Questions :

1- Traduire l'algorithme en Programme PASCAL.

2- Compiler et exécuter le programme pour :

N = 3 et

$$A = \begin{bmatrix} 2 & 4 & -4 \\ 4 & 8 & 1 \\ 3.5 & 9 & 4 \end{bmatrix}$$

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

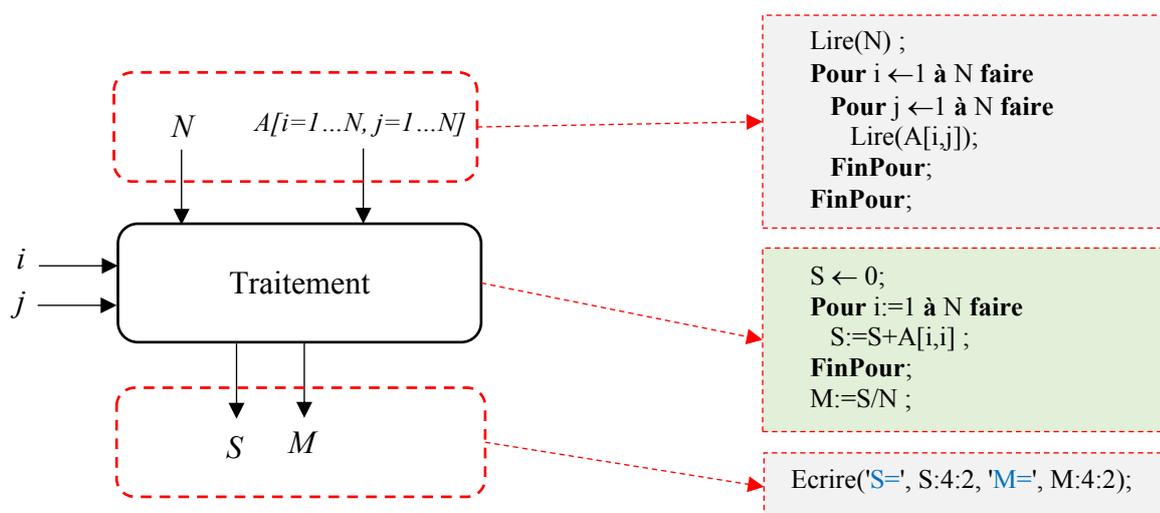
4- Dédire ce que fait le programme ?

5- Ré-écrire le programme en remplaçant la boucle *Pour* par la boucle *Tantque* dans la partie des entrées.

6- Ré-écrire le programme en remplaçant la boucle *Pour* par la boucle *Répéter* dans la partie des entrées.

Solution :

Les variables d'entrée, variable de sortie et la partie traitement sont présentées dans le schéma ci-dessous :



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</p> <p>A : Tableau [1..100, 1..100] de réel;</p> <p>i, j, N : entier;</p> <p>S, M : réel;</p> <p>Début</p> <p style="color: red;">{*-**-* Entrées *-**-*}</p> <p>Ecrire('Donner la taille de la matrice carrée A :');</p> <p>Lire(N);</p> <p>Ecrire('Donner les composantes de la matrice A :');</p> <p>Pour i ← 1 à N faire</p> <p style="padding-left: 20px;">Pour j ← 1 à N faire</p> <p style="padding-left: 40px;">Lire(A[i,j]);</p> <p style="padding-left: 20px;">FinPour;</p> <p>FinPour;</p> <p style="color: red;">{*-**-* Traitements *-**-*}</p> <p>S ← 0;</p> <p>Pour i ← 1 à N faire</p> <p style="padding-left: 20px;">S ← S+A[i,i] ;</p> <p>FinPour;</p> <p>M ← S/N ;</p> <p style="color: red;">{*-**-* Sorties *-**-*}</p> <p>Ecrire('S= ', S:4:2, ' M=', M:4:2);</p> <p>Fin.</p>	<p>Program Matrice ;</p> <p>Var</p> <p>A : array [1..100, 1..100] of real;</p> <p>i, j, N : integer;</p> <p>S, M : real;</p> <p>Begin</p> <p style="color: red;">{*-**-* Entrées *-**-*}</p> <p>Writeln('Donner la taille de la matrice carrée A :');</p> <p>Read(N);</p> <p>Writeln('Donner les composantes de la matrice A :');</p> <p>For i :=1 to N do {La boucle des lignes}</p> <p style="padding-left: 20px;">For j :=1 to N do {La boucle des colonnes}</p> <p style="padding-left: 40px;">Read(A[i,j]);</p> <p style="color: red;">{*-**-* Traitements *-**-*}</p> <p>S := 0;</p> <p>For i:=1 to N do</p> <p style="padding-left: 20px;">S:=S+A[i,i] ;</p> <p style="padding-left: 20px;">M:=S/N ;</p> <p style="color: red;">{*-**-* Sorties *-**-*}</p> <p>Write('S= ', S:4:2, ' M=', M:4:2);</p> <p>End.</p>

2 - Compiler et exécuter le programme pour : N = 3 et

$$A = \begin{bmatrix} 2 & 4 & -4 \\ 4 & 8 & 1 \\ 3.5 & 9 & 4 \end{bmatrix}$$

```

1 Program Matrice ;
2 Var
3   A : array [1..100, 1..100] of real;
4   i, j, N : integer;
5   S, M : real;
6 Begin
7   {*-**-* Entrées *-**-*}
8   Writeln('Donner la taille de la matrice carrée A :');
9   Read(N);
10  Writeln('Donner les composantes de la matrice A :');
11  For i :=1 to N do {La boucle des lignes}
12    For j :=1 to N do {La boucle des colonnes}
13      Read(A[i,j]);
14
15  {*-**-* Traitements *-**-*}
16  S := 0;
17  For i:=1 to N do
18    S:=S+A[i,i] ;
19
20  M:=S/N ;
21  {*-**-* Sorties *-**-*}
22  Write('S= ', S:4:2, ' M=', M:4:2);
23 End.
```

MyPascal V1.20.5 (Exécution)

```

Donner la taille de la matrice carrée A :
3
Donner les composantes de la matrice A :
2 4 -4
4 8 1
3.5 9 4
S= 14.00 M=4.67
```



Après l'exécution

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

Instructions	Variables						Affichage																
	N	i	j	A	S	M																	
Ecrire('Donner la taille de la matrice carrée A :');	/	/	/	/	/	/	Donner la taille de la matrice carrée A :																
Read(N);	3	/	/	/	/	/																	
Ecrire('Donner les composantes de la matrice A :');	3	/	/	/	/	/	Donner les composantes de la matrice A :																
For i:=1 to N do For j:=1 to N do read (A[i,j]) ;	3	1 2 3	1 2 3	<table border="1" style="display: inline-table; vertical-align: middle;"> <tr> <td></td> <td>j=1</td> <td>j=2</td> <td>j=3</td> </tr> <tr> <td>i=1</td> <td>2</td> <td>4</td> <td>-4</td> </tr> <tr> <td>i=2</td> <td>4</td> <td>8</td> <td>1</td> </tr> <tr> <td>i=3</td> <td>3.5</td> <td>9</td> <td>4</td> </tr> </table>		j=1	j=2	j=3	i=1	2	4	-4	i=2	4	8	1	i=3	3.5	9	4	/	/	//
	j=1	j=2	j=3																				
i=1	2	4	-4																				
i=2	4	8	1																				
i=3	3.5	9	4																				
S:=0;	3	/	/	//	0	/	//																
For i:=1 S:=S+A[i,i]; S:=0+A[1,1]; S:=0+2= 2 ;	3	1		//	2	/	//																
For i:=2 S:=S+A[i,i]; S:=2+A[2,2]; S:=2+8= 10 ;	3	2	/	//	10	/	//																
For i:=3 S:=S+A[i,i]; S:=10+A[3,3]; S:=10+4= 14 ;	3	3	/	//	14	/	//																
M:=S/N; M:=14/3= 4.67 ;	3			//	14	4.67	//																
Write(S:4:2, M:4:2);	3			//	14	4.67	S=14 M=4.67																

4 - Dédurre ce que fait le programme ?

Le programme calcul la somme des éléments de la diagonale de la matrice A et leur moyenne.

5 - Ré-écrire le programme en remplaçant la boucle *Pour* par la boucle *Tantque* dans la partie des entrées.

Programme PASCAL (avec la boucle For)	Programme PASCAL (avec la boucle While)
<pre> Program Matrice ; Var A : array [1..100, 1..100] of real; i, j, N : integer; S, M : real; Begin {*-**-* Entrées *-**-*} Writeln('Donner la taille de la matrice carrée A :'); Read(N); Writeln('Donner les composantes de la matrice A :'); For i :=1 to N do {La boucle des lignes} For j :=1 to N do {La boucle des colonnes} Read(A[i,j]); {*-**-* Traitements *-**-*} S := 0; For i:=1 to N do S:=S+A[i,i] ; M:=S/N ; {*-**-* Sorties *-**-*} Write('S= ', S:4:2, ' M=', M:4:2); End. </pre>	<pre> Program Matrice ; Var A : array [1..100, 1..100] of real; i, j, N : integer; S, M : real; Begin {*-**-* Entrées *-**-*} Writeln('Donner la taille de la matrice carrée A :'); Read(N); Writeln('Donner les composantes de la matrice A :'); i:=1 ; While (i<=N) do {La boucle des lignes} Begin j:=1 ; While (j<=N) do {La boucle des colonnes} Begin Read(A[i,j]); j:=j+1; End; i:=i+1; End; {*-**-* Traitements *-**-*} S := 0; For i:=1 to N do S:=S+A[i,i] ; M:=S/N ; {*-**-* Sorties *-**-*} Write('S= ', S:4:2, ' M=', M:4:2); End. </pre>

```

1 Program Matrice ;
2 Var
3   A : array [1..100, 1..100] of real;
4   i, j, N : integer;
5   S, M : real;
6 Begin
7   {*-**-* Entrées *-**-*}
8   Writeln('Donner la taille de la matrice carrée A :');
9   Read(N);
10  Writeln('Donner les composantes de la matrice A :');
11  i:=1 ;
12  While (i<=N) do {La boucle des lignes}
13  Begin
14    j:=1 ;
15    While (j<=N) do {La boucle des colonnes}
16    Begin
17      Read(A[i,j]);
18      j:=j+1;
19    End;
20    i:=i+1;
21  End;
22  {*-**-* Traitements *-**-*}
23  S := 0;
24  For i:=1 to N do
25    S:=S+A[i,i] ;
26  M:=S/N ;
27  {*-**-* Sorties *-**-*}
28  Write('S= ', S:4:2, ' M=', M:4:2);
29 End.

```

MyPascal V1.20.5 (Exécution)

```

Donner la taille de la matrice carrée A :
3
Donner les composantes de la matrice A :
2 4 -4
4 8 1
3.5 9 4
S= 14.00 M=4.67

```



Après l'exécution

6 - Ré-écrire le programme en remplaçant la boucle *Pour* par la boucle *Répéter* dans la partie des entrées.

Programme PASCAL (avec la boucle For)	Programme PASCAL (avec la boucle Repeat)
<pre> Program Matrice ; Var A : array [1..100, 1..100] of real; i, j, N : integer; S, M : real; Begin {*-**-* Entrées *-**-*} Writeln('Donner la taille de la matrice carrée A :'); Read(N); Writeln('Donner les composantes de la matrice A :'); For i :=1 to N do {La boucle des lignes} For j :=1 to N do {La boucle des colonnes} Read(A[i,j]); {*-**-* Traitements *-**-*} S := 0; For i:=1 to N do S:=S+A[i,i] ; M:=S/N ; {*-**-* Sorties *-**-*} Write('S= ', S:4:2, ' M=', M:4:2); End. </pre>	<pre> Program Matrice ; Var A : array [1..100, 1..100] of real; i, j, N : integer; S, M : real; Begin {*-**-* Entrées *-**-*} Writeln('Donner la taille de la matrice carrée A :'); Read(N); Writeln('Donner les composantes de la matrice A :'); i:=1 ; Repeat {La boucle des lignes} j:=1 ; Repeat {La boucle des colonnes} Read(A[i,j]); j:=j+1; Until (j>N); i:=i+1; Until (i>N); {*-**-* Traitements *-**-*} S := 0; For i:=1 to N do S:=S+A[i,i] ; M:=S/N ; {*-**-* Sorties *-**-*} Write('S= ', S:4:2, ' M=', M:4:2); End. </pre>

```

1 Program Matrice ;
2 Var
3   A : array [1..100, 1..100] of real;
4   i, j, N : integer;
5   S, M : real;
6 Begin
7   {*-**-* Entrées *-**-*}
8   Writeln('Donner la taille de la matrice carrée A :');
9   Read(N);
10  Writeln('Donner les composantes de la matrice A :');
11  i:=1 ;
12  Repeat {La boucle des lignes}
13    j:=1 ;
14    Repeat {La boucle des colonnes}
15      Read(A[i,j]);
16      j:=j+1;
17    Until (j>N);
18    i:=i+1;
19  Until (i>N);
20  {*-**-* Traitements *-**-*}
21  S := 0;
22  For i:=1 to N do
23    S:=S+A[i,i] ;
24
25  M:=S/N ;
26  {*-**-* Sorties *-**-*}
27  Write('S= ', S:4:2, ' M=', M:4:2);
28 End.

```

MyPascal V1.20.5 (Exécution)

```

Donner la taille de la matrice carrée A :
3
Donner les composantes de la matrice A :
2 4 -4
4 8 1
3.5 9 4
S= 14.00 M=4.67

```

Après l'exécution

Exercice N°02 : Transposée d'une matrice

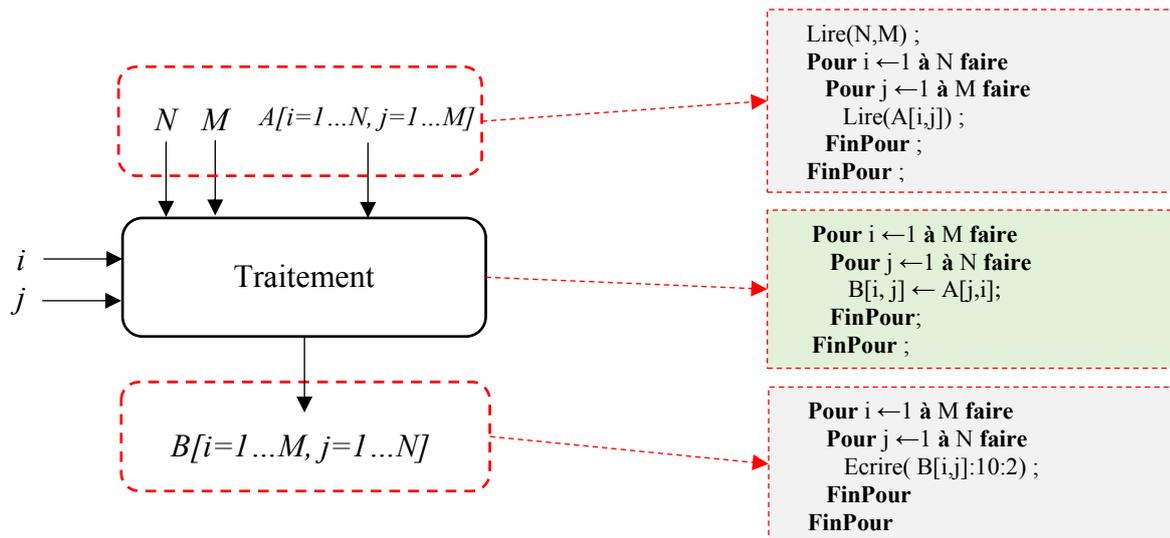
Ecrire un algorithme/programme PASCAL qui permet de calculer la matrice B transposée d'une matrice réelle A d'ordre $N \times M$.

Solution :

Le transposé d'une matrice A d'ordre $N \times M$ est une matrice B d'ordre $M \times N$.

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 : $i=1, \dots, M$ et $j=1, \dots, N$.

Les variables d'entrée, variable de sortie et la partie traitement sont présentées dans le schéma ci-dessous :



Algorithme/programme PASCAL :

Algorithme	Programme PASCAL
<p>Algorithme Transposee ;</p> <p>Variables <i>i, j, N, M</i> : Entier ; <i>A, B</i> : Tableau [1..100, 1..100] de réel ;</p> <p>Début {*--* Entrées *--*} Ecrire('Donner le nombre des lignes et des colonnes de A :'); Lire(N,M); Ecrire('Donner les composantes de la matrice A :'); Pour <i>i</i>←1 à <i>N</i> faire Pour <i>j</i>←1 à <i>M</i> faire Lire(A[i,j]); FinPour ; FinPour ;</p> <p>{*--* Traitements *--*} Pour <i>i</i>←1 à <i>M</i> faire Pour <i>j</i>←1 à <i>N</i> faire <i>B[i, j]</i>← <i>A[j,i]</i> ; FinPour ; FinPour ;</p> <p>{*--* Sorties *--*} Ecrire('La matrice B Transposée de A est : '); Pour <i>i</i>←1 à <i>M</i> faire Pour <i>j</i>←1 à <i>N</i> faire Ecrire(B[i,j]:10:2) ; FinPour ; FinPour ; Fin.</p>	<p>Program Transposee;</p> <p>Var <i>i, j, N, M</i> : integer; <i>A,B</i> : array[1..100, 1..100] of real;</p> <p>Begin {*--* Entrées *--*} Writeln('Donner le nombre des lignes et des colonnes de A :'); Read(N,M); Writeln('Donner les composantes de la matrice A :'); For <i>i:=1 to N do</i> For <i>j:=1 to M do</i> Read(A[i,j]); End;</p> <p>{*--* Traitements *--*} For <i>i:=1 to M do</i> Begin For <i>j:=1 to N do</i> <i>B[i, j] := A[j,i]</i> ; End;</p> <p>{*--* Sorties *--*} Writeln('La matrice B Transposée de A est : '); For <i>i:=1 to M do</i> Begin For <i>j:=1 to N do</i> Write(B[i,j]:10:2) ; {Afficher dans la même ligne} Writeln; {Sauter la ligne} End;</p> <p>End.</p>

```

1 Program Transposee;
2 Var
3   i,j,n,m : integer;
4   A,B : array[1..100, 1..100] of real;
5 Begin
6   {^.*.* Les entrées ^.*.*}
7   Writeln('Donner le nombre des lignes et des colonnes de A :');
8   Read(n,m);
9   Writeln('Donner les composantes de la matrice A :');
10  For i:=1 to n do
11    For j:=1 to m do
12      Read(A[i,j]);
13
14  {^.*.* Traitement ^.*.*}
15  For i:=1 to m do
16    Begin
17      For j:=1 to n do
18        B[i,j] := A[j,i];
19      End;
20
21  {^.*.* Les sorties ^.*.*}
22  Writeln('La matrice B Transposée de A est :');
23  For i:=1 to m do
24    Begin
25      For j:=1 to n do
26        Write(B[i,j]:10:2); {Afficher dans la même ligne}
27        Writeln; {Sauter la ligne}
28      End;
29  End.

```

```

MyPascal V1.20.5 (Exécution) C:\Users\Ahmed\Desktop\...
Donner le nombre des lignes et des colonnes de A :
3 4
Donner les composantes de la matrice A :
2 3 4 5
6 7 8 9
1 2 3 4
La matrice B Transposée de A est :
2.00    6.00    1.00
3.00    7.00    2.00
4.00    8.00    3.00
5.00    9.00    4.00

```

Exercice N°03 : Matrice symétrique

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

Ecrire un programme pascal qui permet de vérifier si la matrice A est symétrique.

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

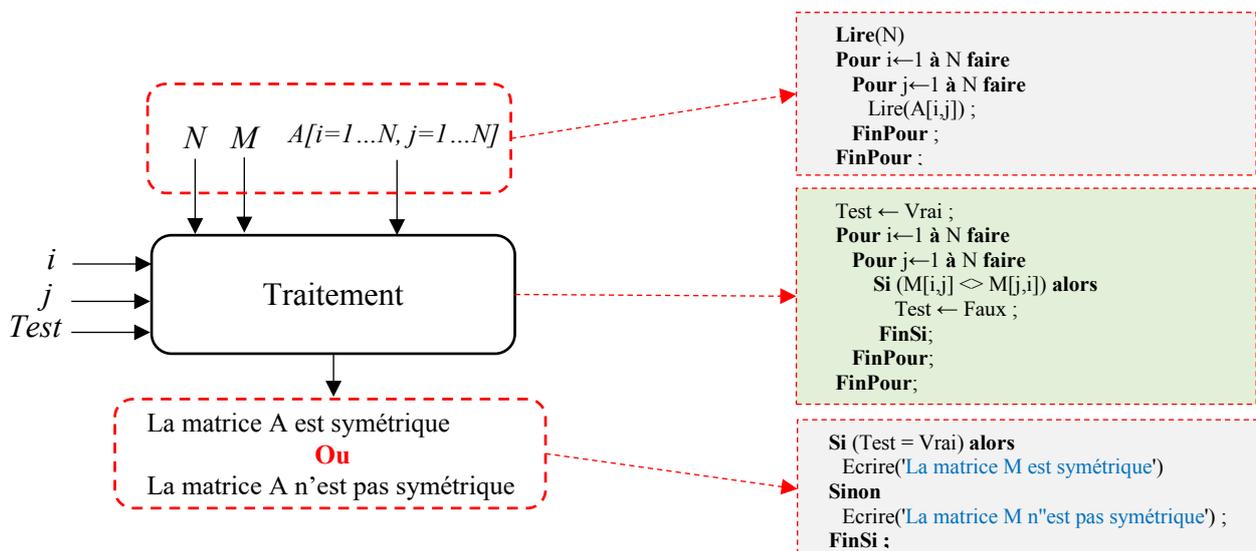
Solution :

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

Les étapes à suivre :

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

Les variables d'entrée, variable de sortie et la partie traitement sont présentées dans le schéma ci-dessous :

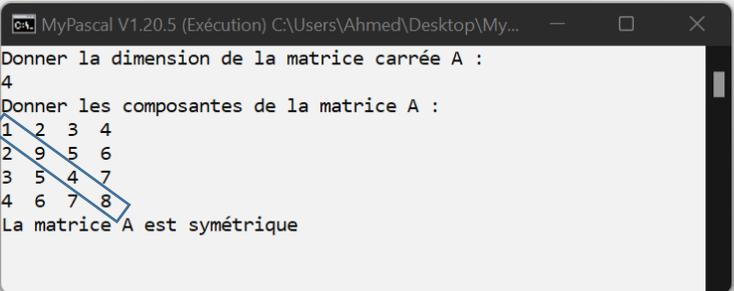


Algorithme/programme PASCAL :

Algorithme	Programme PASCAL
<p>Algorithme Matrice_Symetrique;</p> <p>Variables A : Tableau [1..10,1..10] de réel ; N, i, j : entier ; Test : booléen ;</p> <p>Début {*-.* Entrées *-.*} Ecrire('Donner la dimension de la matrice carrée A :'); Lire(N); Ecrire('Donner les composantes de la matrice A :'); Pour i←1 à N faire Pour j←1 à N faire Lire(A[i, j]); Fin-Pour; Fin-Pour;</p> <p>{*-.* Traitements *-.*} Test ← Vrai ;</p> <p>Pour i←1 à N faire Pour j←1 à N faire Si (A[i,j] <> A[j,i]) alors Test ← Faux ; Fin-Si; Fin-Pour; Fin-Pour;</p> <p>{*-.* Sorties *-.*} Si (Test = Vrai) alors Ecrire('La matrice A est symétrique') Sinon Ecrire('La matrice A n"est pas symétrique') ; Fin-Si ; Fin.</p>	<p>Program Matrice_Symetrique;</p> <p>Var A : array [1..10,1..10] of real ; N, i, j : integer ; Test : boolean ;</p> <p>Begin {*-.* Entrées *-.*} Writeln('Donner la dimension de la matrice carrée A :'); Read(N); Writeln('Donner les composantes de la matrice A :'); For i :=1 to N do For j :=1 to N do Read(A[i, j]);</p> <p>{*-.* Traitements *-.*} Test := True ; For i:=1 to N do Begin For j:=1 to N do if (A[i,j] <> A[j,i]) then Test := False ; End;</p> <p>{*-.* Sorties *-.*} if (Test = True) then Writeln('La matrice A est symétrique') Else Writeln('La matrice A n"est pas symétrique') ; End.</p> <div style="border: 1px dashed black; padding: 5px; margin-top: 10px;"> <p>NB : (begin et end est facultatif) Test := True ; For i:=1 to N do For j:=1 to N do if (A[i,j] <> A[j,i]) then Test := False ;</p> </div>

```

1 Program Matrice_Symetrique;
2 Var
3   A : array [1..10,1..10] of real;
4   N,i,j : integer ;
5   Test : boolean ;
6
7 Begin
8   {*-.* Les entrée *-.*}
9   Writeln('Donner la dimension de la matrice carrée A :');
10  Read(N);
11  Writeln('Donner les composantes de la matrice A :');
12  For i :=1 to N do
13    For j :=1 to N do
14      Read(A[i, j]);
15
16  {*-.* Traitement *-.*}
17  Test := True ;
18  For i:=1 to N do
19    Begin
20      For j:=1 to N do
21        if (A[i,j] <> A[j,i]) then
22          Test := False ;
23    End;
24
25  {*-.* Les sorties *-.*}
26  if (Test = True) then
27    Writeln('La matrice A est symétrique')
28  Else
29    Writeln('La matrice A n"est pas symétrique') ;
30 End.
```




 Après l'exécution

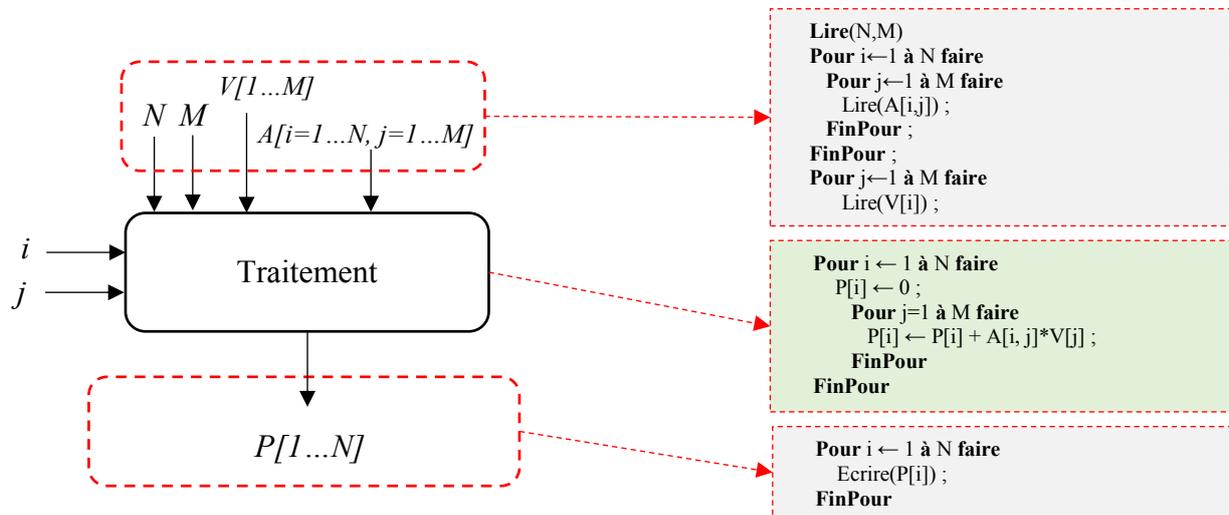
Exercice N°04 : Produit d'une matrice par un vecteur

Soit A une matrice de type réel et d'ordre $N \times M$.

Ecrire un algorithme/programme PASCAL qui permet de calculer le produit de la matrice A par un vecteur V de type réel et de taille M.

Solution :

Les variables d'entrée, variable de sortie et la partie traitement sont présentées dans le schéma ci-dessous :



Algorithme/programme PASCAL :

Algorithme	Programme PASCAL
<p>Algorithme Produit-Matrice_Vecteur ;</p> <p>Variabes A : Tableau [1..10, 1..10] de réel ; V, P : Tableau [1..10] de réel ; N, M, i, j : entier ;</p> <p>Début</p> <pre> {*-*-* Entrées *-*-*} Ecrire('Donner la dimension de la matrice carrée A : '); Lire (N, M); Ecrire('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 ; Ecrire('Donner les composantes du vecteur V : '); pour i ← 1 à M faire Lire(V[i]) ; {*-*-* Traitements *-*-*} FinPour ; pour i ← 1 à N faire P[i] ← 0 ; Pour j=1 à M faire P[i] ← P[i] + A[i, j]*V[j] ; FinPour; FinPour ; {*-*-* Sorties *-*-*} Ecrire('Le résultat de produit :'); pour i ← 1 à N faire Ecrire(P[i]:8:2) ; FinPour Fin. </pre>	<p>Program Produit_Matrice_Vecteur;</p> <p>Var A : array [1..10, 1..10] of real ; V, P : array [1..10] of real ; N, M, i, j : integer ;</p> <p>Begin</p> <pre> {*-*-* Entrées *-*-*} Write('Donner la dimension de la matrice carrée 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]); Writeln('Donner les composantes du vecteur V : '); For i:=1 to M do Read(V[i]); {*-*-* Traitements *-*-*} For i:=1 to N do Begin P[i]:=0; For j:=1 to M do P[i]:= P[i]+ A[i,j]*V[j]; End; {*-*-* Sorties *-*-*} Writeln('Le résultat de produit :'); For i:=1 to N do write(P[i]:8:2); End. </pre>

```

1 Program Produit_Matrice_Vecteur;
2 Var
3   A : array [1..10, 1..10] of real;
4   V, P : array [1..10] of real;
5   N, M, ij : integer;
6 Begin
7   {*** Entrées ***}
8   Write('Donner la dimension de la matrice carrée A : ');
9   Read(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  Writeln('Donner les composantes du vecteur V : ');
15  For i:=1 to M do
16    Read(V[i]);
17  {*** Traitements ***}
18  For i:=1 to N do
19    Begin
20      P[i]:=0;
21      For j:=1 to M do
22        P[i]:= P[i]+ A[i,j]*V[j];
23    End;
24  {*** Sorties ***}
25  Writeln('Le résultat de produit :');
26  For i:=1 to N do
27    write(P[i]:8:2);
28 End.

```

```

MyPascal V1.20.5 (Exécution) C:\Users\Ahmed\Desktop\Pr...
Donner la dimension de la matrice carrée A : 3 4
Donner les composantes de la matrice A :
1 2 3 4
5 4 3 6
8 7 4 1
Donner les composantes du vecteur V :
3 5 7 1
Le résultat de produit :
38.00 62.00 88.00

```

