

**EXAMEN DE REMPLACEMENT –
 INFORMATIQUE 2**

Durée: 01h30min

Questions de cours : [04 points]

1. Quelle est la différence entre une matrice carrée et une matrice rectangulaire ?
2. Donner deux syntaxes différentes pour déclarer la variable A , tel que A est une matrice de taille (10×50) de type réel ?
3. Citez deux conditions nécessaires pour une transmission correcte entre les paramètres effectifs et les paramètres formels ?
4. Donner les modifications à faire pour transformer une fonction en une procédure.

Exercice 01 : [10 points]

Partie A [06 points] : Soit l'algorithme suivant :

Algorithme partie_A;

Variables

T : Tableau [1..100] d'entier ;

N, i, j, temp : entier ;

Début

Lire(N) ;

Pour i ← 1 à N **faire**

 Lire(T[i]) ;

Fin-Pour;

Pour i ← 1 à N-1 **faire**

Pour j ← i+1 à N **faire**

Si (T[i] > T[j]) **alors**

 temp ← T[i] ;

 T[i] ← T[j] ;

 T[j] ← temp ;

Fin-Si;

Fin-Pour;

Fin-Pour;

Pour i ← 1 à N **faire**

 Écrire(T[i] :3) ;

Fin-Pour ;

Fin.

Questions :

1. Traduire l'algorithme en programme PASCAL.
2. Dérouler l'algorithme pour T=

1	0	2	-1
---	---	---	----
3. Déduire ce que fait l'algorithme.
4. Quel est le rôle de la variable temp ?
5. Réécrire la **partie sortie** du programme en utilisant la boucle **Répéter**.

Partie B [04 points] :

Soit A une matrice de taille $(N \times M)$ contenant des entiers. On considère T la matrice transposée de A , et on définit une nouvelle matrice B comme étant le produit du coefficient entier x par la matrice transposée T , soit $B = x \times T$. Écrire un programme en langage Pascal qui permet de lire les éléments de la matrice A , de calculer sa transposée T , puis de calculer la matrice B , et enfin d'afficher les éléments de cette matrice B .

Exercice 02 : [6 points]

Soit l'expression suivante : $M = \prod_{i=1}^N (X!)$, écrire un programme PASCAL qui:

- Contient un sous-programme (fonction ou procédure) nommé **Fact** afin de calculer la factorielle.
- Calcule M en appelant le sous-programme **Fact** et affiche le résultat de M .

**CORRIGÉ DE L'EXAMEN DE
 REMPLACEMENT-INFORMATIQUE 2**

Durée: 01h30min

Questions de cours : [04 points]

1. Quelle est la différence entre une matrice carrée et une matrice rectangulaire ? **(01 Pts)**

Matrice carrée	Matrice rectangulaire
- Lignes = Colonnes (ex. : 4×4)	- Lignes ≠ Colonnes (ex. : 3×5, 10×2)
- Possède déterminant, inverse, etc.	- Ne possède pas toujours ces propriétés

0.5*2

2. Donner deux syntaxes différentes pour déclarer la variable A, telle que A est une matrice de taille (10*50) de type réel ? **(01 Pts)**

1- A : array [1..10,1..50] of real;

0.5

2- Type tab :array [1..10] of real ;
 A: array [1..50] of tab;

0.5

3- Citez deux conditions nécessaires pour une transmission correcte entre les paramètres effectifs et les paramètres formels ? **(01 Pts)**

- | | |
|----------------------------|------------------------------|
| - Correspondance en nombre | - Ordre respecté |
| - Correspondance en type | - Mode de passage compatible |

0.5*2

4- Donner les modifications à faire pour transformer une fonction en une procédure. **(01 Pts)**

- | |
|---|
| - Remplacer fonction par procédure - Supprimer le type de retour |
| - Ajouter un paramètre de sortie avec var – supprimé la ligne d'affectation du résultat (ex. Nom_fonction):=...) |

0.5*2

Exercice 01 : [10points]

Partie A (06 Pts):

1. Traduire l'algorithme en programme PASCAL ; **(02 pts)**

<p>Algorithme partie_A;</p> <p>Variables</p> <p>T : Tableau [1..100] d'entier ;</p> <p>N, i, j, temp : entier ;</p> <p>Début</p> <p> Lire(N) ;</p> <p> Pour i ← 1 à N faire</p> <p> Lire(T[i]) ;</p> <p> Fin-Pour;</p> <p> Pour i ← 1 à N-1 faire</p> <p> Pour j ← i+1 à N faire</p> <p> Si (T[i] >T[j]) alors</p> <p> temp ← T[i] ;</p> <p> T[i] ← T[j] ;</p> <p> T[j] ← temp ;</p> <p> Fin-Si;</p> <p> Fin-Pour;</p> <p> Fin-Pour;</p> <p> Pour i ← 1 à N faire</p> <p> Écrire(T[i] :3) ;</p> <p> Fin-Pour ;</p> <p>Fin.</p>	<p>Program partie_A ; 0.125</p> <p> Var</p> <p> T : array [1..10] of integer ; } 0.25</p> <p> N, i, j, temp : integer ;</p> <p> Begin</p> <p> Read (N); 0.125</p> <p> For i:=1 to N do 0.125</p> <p> Read (T[i]); 0.125</p> <p> For i:=1 to N-1 do 0.125</p> <p> For j:=i+1 to N do 0.125</p> <p> If T[i] > T[j] then 0.125</p> <p> Begin</p> <p> temp := T[i] ; 0.125</p> <p> T[i] := T[j] ; 0.125</p> <p> T[j] := temp ; 0.125</p> <p> End ;</p> <p> For i:=1 to N do } 0.25</p> <p> Write (T[i]:3);</p> <p> End.</p>
--	--

2. Dérouler l'algorithme (**02 pts**)

Instructions	Variables					Affichage
	N	i	j	temp	T	
Lire (N) ; 0.25	4	-	-	-	-	-
Pour i ← 1 à N faire Lire(T[i]) ;	4	1 2 3 4	-	-	1 0 2 -1 0.25	-
Pour i ← 1 Pour j ← 2 Si T[1] > T[2] True temp ← T[1] ; T[1] ← T[2] ; 0.125 T[2] ← temp ; Pour j ← 3 0.125 Si T[1] > T[3] False Pour j ← 4 Si T[1] > T[4] True temp ← T[1] ; 0.125 T[1] ← T[4] ; T[4] ← temp ;	4	1	2 3 4	1 0.125 0 0.125	0 1 2 -1 -1 1 2 0	-
Pour i ← 2 Pour j ← 3 Si T[2] > T[3] False Pour j ← 4 0.125 Si T[2] > T[4] True temp ← T[2] ; 0.125 T[2] ← T[4] ; T[4] ← temp ;	4	2	3 4	1 0.125	-1 0 2 1	-
Pour i ← 3 Pour j ← 4 Si T[3] > T[4] True temp ← T[3] ; 0.125 T[3] ← T[4] ; T[4] ← temp ;	4	3	4	2 0.125	-1 0 1 2	-
Pour i ← 1 à N faire Écrire(T[i] :3) ;	4	1 2 3 4	4	2	-1 0 1 2	0.25 -1 0 1 2

3. L'algorithme permet Cet algorithme effectue un tri simple par échange pour ordonner les éléments d'un tableau dans l'ordre croissant. **0.5**

4. temp : joue le rôle d'une variable temporaire qui est utilisée pour échanger les valeurs dans le tableau T **0.5**

5. Réécrire la partie **Sortie** du programme en utilisant la boucle **Répéter**. (01pt)

```

i:=1; 0.25
repeat 0.25
Write (T[i]:3);
i:=i+1; 0.25
until (i>N); 0.25

```

Partie B: (04pts) - Soit A une matrice de taille $(N \times M)$ contenant des entiers. On considère T la matrice transposée de A , et on définit une nouvelle matrice B comme étant le produit du coefficient entier x par la matrice transposée T , soit $B = x \times T$. Écrire un programme en langage Pascal qui permet de lire les éléments de la matrice A , de calculer sa transposée T , puis de calculer la matrice B , et enfin d'afficher les éléments de cette matrice B .

Programme en Pascal	
	<pre> Program partie_B; Var A, T, B : array [1..10,1..10] of integer; N, M, X, i, j : integer; Begin write('Donner N et M : '); readln(N, M); writeln('Donner les composantes de A :'); for i := 1 to N do for j := 1 to M do read(A[i, j]); for i := 1 to N do for j := 1 to M do T[j, i] := A[i, j]; writeln('La matrice transposée de A est :'); for i := 1 to M do begin for j := 1 to N do write(T[i, j]:5); writeln; end; write('Donner le coefficient x : '); readln(x); for i := 1 to M do for j := 1 to N do B[i, j] := x * T[i, j]; writeln('La matrice B est :'); for i := 1 to M do begin for j := 1 to N do write(B[i, j]:5); writeln; end; End. </pre>
0.25	A, T, B : array [1..10,1..10] of integer;
N, M, X, i, j : integer;	
0.5	Begin
write('Donner N et M : ');	
readln(N, M);	
writeln('Donner les composantes de A :');	
0.375	for i := 1 to N do
for j := 1 to M do	
read(A[i, j]);	
0.5	for i := 1 to N do
for j := 1 to M do	
T[j, i] := A[i, j];	
writeln('La matrice transposée de A est :');	
0.75	for i := 1 to M do
begin	
for j := 1 to N do	
write(T[i, j]:5);	
writeln;	
end;	
0.25	write('Donner le coefficient x : ');
readln(x);	
0.5	for i := 1 to M do
for j := 1 to N do	
B[i, j] := x * T[i, j];	
writeln('La matrice B est :');	
0.75	for i := 1 to M do
begin	
for j := 1 to N do	
write(B[i, j]:5);	
writeln;	
end;	
0.125	End.

Exercice 02 :[06 points]

Soit l'expression suivante : $M = \prod_{i=1}^N (X!)$. Ecrire un programme PASCAL qui:

- Contient un sous-programme (fonction ou procédure) nommé **Fact** afin de calculer la factorielle.
- Calcule **M** en appelant le sous-programme Fact et affiche le résultat **M**.

Programme avec une fonction	Programme avec une procédure
<pre> Program exo2 ; <u>0.25</u> Var M,N, i, x : integer ; <u>0.5</u> function Fact(k:integer):integer; <u>0.5</u> var j,f:integer; <u>0.5</u> Begin f:=1; <u>0.5</u> <u>0.25</u> for j:=1 to k do <u>0.25</u> f:=f*j; <u>0.5</u> Fact:=f; <u>0.5</u> end; Begin write ('donner N '); Read (N); <u>0.25</u> write ('donner x '); <u>0.25</u> Read (x); <u>0.25</u> M:=1; <u>0.25</u> For i:=1 to N do <u>0.25</u> M:=M*Fact(x); <u>0.5</u> write ('M= ',M); <u>0.5</u> end. </pre>	<pre> Program exo2 ; <u>0.25</u> Var M,N, i, x,f : integer ; <u>0.5</u> procedure Fact(k:integer; var f : integer); <u>0.5</u> var j:integer; <u>0.5</u> Begin f:=1; <u>0.5</u> <u>0.25</u> for j:=1 to k do <u>0.25</u> f:=f*j; <u>0.5</u> end; Begin write ('donner N '); Read (N); <u>0.25</u> write ('donner x '); Read (x); <u>0.25</u> <u>0.25</u> M:=1; <u>0.25</u> Fact(x,f); <u>0.5</u> For i:=1 to N do <u>0.25</u> M:=M*f; <u>0.5</u> write ('M=',M); <u>0.5</u> end. </pre>