

Corrigé de l'Examen de Rattrapage – Informatique 2

Exercice 01 : [3 pts]

- 1) Citer les deux types de variables indicées. (1 point)
 - Vecteur ou tableau à une dimension (0.5 point)
 - Matrice ou tableau à deux dimensions (0.5 point)
- 2) Citer les deux types de sous-programmes. (1 point)
 - Fonction (0.5 point)
 - Procédure (0.5 point)
- 3) Qu'est-ce qu'une variable locale dans un programme contenant un sous-programme ? (1 point)
 - C'est une variable qui est définie dans un sous-programme, (0.5 point)
 - C'est une variable qui est inaccessible par le programme principal et par d'autres sous-programmes. (0.5 point)

Exercice 02 : [10 pts]

Partie A : [6 pts]

Soit l'algorithme suivant :

Algorithme Exercice2_A

Variables

V : tableau [1..50] de réel

N, i : entier

Z : réel

Début

Lire(N) ;

Pour i ← 1 à N faire

Lire(V[i])

Fin-pour

Pour i ← 1 à (N div 2) faire

Z ← V[i]

V[i] ← V[N - i + 1]

V[N - i + 1] ← Z

Fin-pour

Pour i ← 1 à N faire

Écrire(V[i])

Fin-pour

Fin

Questions :

- 1) Quelles sont les variables d'entrée, intermédiaires et de sortie de l'algorithme ? (1.25 point)
- 2) Traduire l'algorithme en programme Pascal. (2 points)
- 3) Dérouler l'algorithme pour : (2.25 points)
$$N = 5 \quad V = \begin{array}{|c|c|c|c|c|} \hline 4 & 0 & 8 & 2 & 1 \\ \hline \end{array}$$
- 4) Déduire ce que fait l'algorithme. (0.5 point)

- 1) Quelles sont les variables d'entrée, intermédiaires et de sortie de l'algorithme ? (1.25 point)

Les variables d'entrée : N et V (les cases de 1 à N) (0.25 × 2 = 0.5 point)

Les variables intermédiaires : i et Z (0.25 × 2 = 0.5 point)

Les variables de sortie : V (les cases de 1 à N) (0.25 point)

- 2) Traduire l'algorithme en programme Pascal. (2 points)

Program Exercice2_A; (0.25 point)

Var V : Array [1..50] of real ; (0.25 point)

N, i : integer ; Z : real ; (0.25 point)

BEGIN

Read(N) ; (0.25 point)

For i := 1 **to** N **do**
Read (V[i]) ; } (0.25 point)

For i := 1 **to** (N div 2) **do** (0.125 point)
Begin (0.125 point)
Z := V[i] ;
V[i] := V[N - i + 1] ;
V[N - i + 1] := Z ; } (0.125 point) } (0.5 point)
End ; (0.125 point)

For i := 1 **to** N **do**
Write (V[i]) ; } (0.25 point)

END.

3) Dérouler l'algorithme pour N = 5 et V =

4	0	8	2	1
---	---	---	---	---

 (2.25 points)

Instructions	Variables				Affichage										
	N	i	V	Z											
Lire(N)	5	/	/	/											
<u>Pour</u> i ← 1 à N faire avec N=5 Lire(V[i]) <u>Fin-pour</u>		1 2 3 4 5	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>4</td><td>0</td><td>8</td><td>2</td><td>1</td></tr></table>	4	0	8	2	1	/						
4	0	8	2	1											
<u>Pour</u> i ← 1 à (N div 2) faire avec N = 5 → 5 div 2 = 2 <u>Pour</u> i ← 1 Z ← V[i] Z ← V[1] Z ← 4 V[i] ← V[N - i + 1] V[1] ← V[5 - 1 + 1] V[1] ← V[5] V[1] ← 1 V[N - i + 1] ← Z V[5 - 1 + 1] ← Z V[5] ← Z V[5] ← 4		1	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>1</td><td>0</td><td>8</td><td>2</td><td>1</td></tr></table> <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>1</td><td>0</td><td>8</td><td>2</td><td>4</td></tr></table>	1	0	8	2	1	1	0	8	2	4	4	
1	0	8	2	1											
1	0	8	2	4											
<u>Pour</u> i ← 2 Z ← V[i] Z ← V[2] Z ← 0 V[i] ← V[N - i + 1] V[2] ← V[5 - 2 + 1] V[2] ← V[4] V[2] ← 2 V[N - i + 1] ← Z V[5 - 2 + 1] ← Z V[4] ← Z V[4] ← 0		2	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>1</td><td>2</td><td>8</td><td>2</td><td>4</td></tr></table> <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>1</td><td>2</td><td>8</td><td>0</td><td>4</td></tr></table>	1	2	8	2	4	1	2	8	0	4	0	
1	2	8	2	4											
1	2	8	0	4											
<u>Pour</u> i ← 1 à N faire avec N=5 Écrire(V[i]) <u>Fin-pour</u>	5	1 2 3 4 5	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>1</td><td>2</td><td>8</td><td>0</td><td>4</td></tr></table>	1	2	8	0	4	0	1 2 8 0 4					
1	2	8	0	4											

4) Dédurre ce que fait l'algorithme. (0.5 point)

L'algorithme permet d'inverser les éléments d'un vecteur V (0.25 point)
dans le même vecteur V (0.25 point)

Partie B : [4 pts]

Soit A une matrice d'ordre $N \times M$ et de type entier. Écrire **un seul programme Pascal** à nommer « Exercice2_B » pour :

- 1) Calculer et afficher le produit des éléments de la dernière ligne de la matrice A (ligne N).
- 2) Calculer et afficher la somme de chaque colonne de la matrice A.

```
Program Exercice2_B ; } (0.25 point)
Var
A : Array [1..50, 1..50] of integer ; (0.25 point)
N, M, i, j : integer ; (0.25 point)
P : integer ; (0.25 point)
Som : Array [1..50] of integer ; (0.25 point)

BEGIN
  Read(N, M) ;
  For i := 1 to N do
    For j := 1 to M do
      Read ( A[i, j] ) ; } (0.25 point)

  P := 1 ; (0.25 point)
  For j := 1 to M do (0.25 point)
    P := P * A[N, j] ; (0.25 point) } (0.75 point)

  Write ( 'Le produit des éléments de la dernière ligne de A est : ', P ) ; (0.25 point)

  For j := 1 to M do (0.25 point)
    Begin (0.125 point)
      Som[j] := 0 ; (0.25 point)
      For i := 1 to N do (0.25 point)
        Som[j] := Som[j] + A[i, j] ; (0.25 point)
      End ; (0.125 point) } 1.25 point

  For j := 1 to M do } (0.25 point)
    Write ( Som[j] ) ;

END.
```

Exercice 03 : [7 pts]

Soit le programme Pascal suivant :

```
Program Exercice3;
Var X, A : Integer;
..... Sous_Prog (N : Integer) : Integer;
Begin
  If N < 0 then
    N := - N;
    .....
End;
Begin {Début du programme principal}
  Read(X);
  ... := Sous_Prog(...);
  Write(A);
End. {Fin du programme principal}
```

Questions :

- 1) Compléter le programme Pascal. **(1.5 point)**
- 2) Quels sont les paramètres formels ? effectifs ? **(0.5 point)**
- 3) Quel est le type du résultat de la fonction ? **(0.5 point)**
- 4) Dérouler le programme pour $X = - 5$. **(2.5 points)**
- 5) Dédire ce que fait le programme. **(0.5 point)**
- 6) Réécrire le programme en transformant la fonction en une procédure de même nom. **(1.5 point)**

- 1) Compléter le programme Pascal. **(1.5 point)**

```
Program Exercice3;
Var X, A : Integer;
Function Sous_Prog (N : Integer) : Integer; (0.5 point)

Begin
  If N < 0 then
    N := - N;
    Sous_Prog := N ; (0.5 point)
End;
Begin {Début du programme principal}
  Read(X);
  A := Sous_Prog( X ); (0.5 point)
  Write(A);
End. {Fin du programme principal}
```

- 2) Quels sont les paramètres formels ? effectifs ? **(0.5 point)**
 - Les paramètres formels : **N (0.25 point)**
 - Les paramètres effectifs : **X (0.25 point)**
- 3) Quel est le type du résultat de la fonction ? **(0.5 point)**
Le type de la fonction est **Entier ou integer (0.5 point)**

4) Dérouler le programme pour X = - 5. (2.5 points)

Instructions	Variables du PP		Variables du SP		Affichage
	X	A	N	Sous_Prog	
Read(X)	- 5	/	/	/	
A := Sous_Prog(X); A := Sous_Prog(-5) Appel à la fonction Sous_Prog et transmission de paramètres		A = ?	-5		
If N < 0 If -5 < 0 True On exécute le bloc If N := -N N := 5 Sous_Prog := N Sous_Prog := 5			5		
A := Sous_Prog(-5) A := 5		5			
Write (A)	-5	5	5	5	5

5) Déduire ce que fait le programme. (0.5 point)

Le programme calcule la valeur absolue d'un nombre entier X (0.25 point) en utilisant la fonction Sous_Prog. (0.25 point)

6) Réécrire le programme en transformant la fonction en une procédure de même nom. (1.5 point)

Solution N°1	Solution N°2
Program Exercice3; Var X, A : Integer; Procedure Sous_Prog (N : Integer; Var M : Integer) : Integer; Begin If N < 0 then M := - N ; Else M := N ; Sous_Prog := N ; End ; Begin {Début du programme principal} Read(X); Sous_Prog(X, A); Write(A); End. {Fin du programme principal}	Program Exercice3; Var X : integer; A : Integer; Procedure Sous_Prog (Var N : Integer) : Integer; Begin If N < 0 then N := - N ; Sous_Prog := N ; End ; Begin {Début du programme principal} Read(X); Sous_Prog(X); Write(X); End. {Fin du programme principal}