

Examen de remplacement – Informatique 2

Questions de cours : [5 pts]

1. En algorithmique, les variables de type entier, réel, etc. sont appelées variables simples, comment appelle-t-on, donc, les variables de type tableau ?
2. Quelle structure permet de lire et d'afficher un tableau ?
3. Comment reconnaît-on un paramètre formel ?
4. Pour être valide, les paramètres d'appel d'un sous-programme doivent respecter deux conditions, lesquelles ?
5. Citez les différents types de passage (transmission) de paramètres.

Exercice 01 : [8 pts]

Soit A une matrice de N x M composantes réelles et T un vecteur de N composantes entières :

- Écrire un algorithme qui permet de trouver la valeur minimum positive (le minimum entre les valeurs positives) et la valeur maximum négative (le maximum entre les valeurs négatives) de la matrice A.

N.B : On considère que la première composante est positive et que la deuxième est négative.

- Remplacer :
 - Les composantes de la première colonne de la matrice A par la racine carrée des composantes du vecteur T (seulement si la composante est positive).
 - Les composantes de la dernière colonne de la matrice A par le carré des composantes du vecteur T.

Exercice 02 : [7 pts]

Soit le programme PASCAL suivant :

```
Program Exo2;
var
a ,b, c: integer;
..... Conversion1 (x : integer; var m : .....);

Begin
m := x div 60;
End;
..... Conversion2 (x : integer; var r : .....);

Begin
r := x mod 60;
End;

Begin
read(a);
Conversion1(a, b);
Conversion2(a,c);
Writeln(a,' secondes = ', b,' minutes et ',c,
' secondes');
End.
```

Questions :

1. Identifier le type de sous-programme présent dans ce programme.
2. Réécrire le sous-programme en complétant avec ce qui manque.
3. Dérouler le programme pour a=31251.
4. Déduire ce que fait le programme ?
5. Réécrire le programme en transformant le sous-programme utilisé à l'autre type de sous-programme.

Bonne chance

Examen de remplacement – Informatique 2

Questions de cours : [5 pts]

1. En algorithmique, les variables de type entier, réel, etc. sont appelées variables simples, comment appelle-t-on, donc, les variables de type tableau ?
 - Variables indicées. (1pt)
2. Quelle structure permet de lire et d'afficher un tableau ?
 - Les boucles. (1 pt)
3. Comment reconnaît-on un paramètre formel ?
 - Les paramètres formels sont les paramètres utilisés lors de la déclaration des sous-programmes. (1 pt)
4. Pour être valide, les paramètres d'appel d'un sous-programme doivent respecter deux conditions, lesquelles ?

Les paramètres d'appel doivent :

 - Être de même nombre que les paramètres formels (0.5 pt)
 - Avoir, respectivement, le même type (0.5 pt)
5. Citez les différents types de passage (transmission) de paramètres. (Deux réponses suffisent)
 - Par valeur. (0.5 pt).
 - Par variable. (0.5 pt).
 - Par fonction.
 - Par procédure.

Exercice 01 : [8 pts]

- 1- Un algorithme qui permet de trouver la valeur minimum positive (le minimum entre les valeurs positives) et la valeur maximum négative (le maximum entre les valeurs négatives) de la matrice A. (5pts)
- 2- Remplacer : (3pts)
 - Les composantes de la première colonne de la matrice A par la racine carrée des composantes du vecteur T (seulement si la composante est positive).
 - Les composantes de la dernière colonne de la matrice A par le carré des composantes du vecteur T.

Algorithme

Algorithme exercice_1 ; } 0.25

Variables

A : Tableau [1..100,1..100] de réel ; } 0.25

T : Tableau [1..100] d'entier ; } 0.25

N,M,i,j : entier ; } 0.25

MP,MN : réel ; } 0.25

Début

Lire (N,M) ; } 0.25

Pour i ← 1 à N faire
 Pour J ← 1 à M faire
 Lire (A[i,j]) ;
 FinPour ;
FinPour ; } 0.5

Pour i ← 1 à N faire
 Lire (T[i]) ;
FinPour ; } 0.25

MP ← A[1,1] ; } 0.25

MN ← A[1,2] ; } 0.25

Pour i ← 1 à N faire
 Pour j ← 1 à M faire } 0.25
 Si ((A[i,j] > 0) et (A[i,j] < MP)) Alors } 0.5
 MP ← A[i,j] ; } 0.25
 Sinon

 Si ((A[i,j] < 0) et (A[i,j] > MN)) Alors } 0.5
 MN ← A[i,j] ; } 0.25

 FinSi ;

 FinPour ;

FinPour ;

Ecrire ('le minimum entre les valeurs positives = ', MP) ; } 0.25

Ecrire ('le maximum entre les valeurs négatives = ', MN) ; } 0.25

```

Pour i ← 1 à N faire } 0.25
  Si (T[i] > 0) Alors } 0.5
    A[i,1] ← Sqrt (T[i]) ; } 0.75
  FinSi ;
  A[i,M] ← Sqr (T[i]) ; } 0.75
  {ou bien A[i,M] ← T[i]* T[i]}

```

FinPour ;

```

Pour i ← 1 à N faire } 0.5
  Pour J ← 1 à M faire } 0.25
    Ecrire (A[i,j]) ; }
  FinPour ;

```

FinPour ;

Fin.

Ou bien

```

Pour i ← 1 à N faire
  Si (T[i] > 0) Alors
    A[i,1] ← Sqrt (T[i]) ;
  FinSi ;
FinPour ;

```

```

Pour i ← 1 à N faire
  A[i,M] ← Sqr (T[i]) ;
FinPour ;

```

Exercice 02 : [7 pts]

1- Identifier le type de sous-programme présent dans ce programme.

Le sous-programme est une procédure. (0,25 pt)

2. Réécrire le sous-programme en complétant avec ce qui manque. (1pt)

Programme Pascal

Program Exo2;

var a ,b, c: integer;

Procedure Conversion1 (x : integer; var m : integer); (0,25+0.25)

Begin

m := x div 60;

End;

Procedure Conversion2 (x : integer; var r : integer); (0,25+0.25)

Begin

r := x mod 60;

End;

Begin

read(a);

Conversion1(a, b);

Conversion2(a,c);

Writeln (a,' secondes = ', b,' minutes et ',c, ' secondes');

End.

3. Dérouler le programme pour a=31251. (2,5 pts)

Instructions	Programme principale			Procédure 1		Procédure 2		Affichage
	a	b	c	x	m	x	r	
read(a);	31251	/	/					
Conversion1(a, b); (L'appel à la procédure conversion1 avec le paramètre a = 31251).								
=> La transmission des paramètres m := x div 60;		520		31251	520			
Conversion2(a, c); (L'appel à la procédure conversion1 avec le paramètre a = 31251).								
=> La transmission des paramètres r := x mod 60;			51			31251	51	
Writeln (a,' secondes = ', b,' minutes et ',c, ' secondes');								31251 secondes = 520 minutes et 51 secondes

4. Déduire ce que fait le programme ? (0.75pt)

Le programme permet de faire la **conversion** des **secondes** en **minutes**.

5. Réécrire le programme en transformant le type de sous-programme utilisé à l'autre type de sous-programme. (2.5 pts)

Programme Pascal	
Program Exo2; var a ,b, c: integer; Function Conversion1 (x : integer) : integer; } Var m : integer ; } Begin m := x div 60; } Conversion1 := m } End; }	
	0.25 pour Function 0.25 pour le type 0.25 pour var m :integer 0.25 pour conversion1 := m
Function Conversion2 (x : integer) : integer; Var r : integer ; Begin r := x mod 60; Conversion2 := r ; End;	
	0.25 pour Function 0.25 pour le type 0.25 pour var r :integer 0.25 pour conversion2 := m
Begin Read(a); b := Conversion1(a);	
	0.25

```
c := Conversion2(a); 0.25
```

```
Writeln (a, ' secondes = ', b, ' minutes et ', c, ' secondes');
```

```
End.
```