



SECTIONS : MATHÉMATIQUES, SCIENCES EXPÉRIMENTALES ET SCIENCES TECHNIQUES

Nom et prénom : Classe : 4^{ème} N° :.....

Partie I : 8 Points

Exercice N°1 : 4 Points

Soit le tableau PAT qui donne le nombre de patients ayant visités un certain cabinet médical par mois :

292	300	231	182	169	143	151	128	156	169	192	251
Janvier	Février	Mars	Avril	Mai	Juin	Juillet	Août	Septembre	Octobre	Novembre	Décembre

Questions :

1) En utilisant le type scalaire énuméré, déclarer en Pascal un type Mois contenant les mois de l'année.

.....

2) Soit TAB le type du tableau PAT, déclarer en Pascal le type TAB.

.....

3) Compléter le tableau ci-dessous en mettant une croix (X) dans la colonne valide ou dans la colonne non valide pour chaque instruction. Justifier votre réponse pour les instructions non valides.

Déclarations / instructions	Valide	Non valide	Justification
Writeln(PAT[Mai]);			
Var i :string ; begin For i := Janvier to Décembre do PAT[i]:=2*PAT[i];			
Function Max(p:TAB):Mois;			
Var M:Mois; begin Readln(M);			

Exercice N°2 : 4 Points

Un entier n ($n >= 10$) est **divisible par 13**, si on multiplie successivement le dernier chiffre par 4 et on l'additionne au nombre privé de son dernier chiffre, le traitement s'arrête lorsque le résultat obtenu est composé de 2 chiffres, s'il est un multiple de 13, alors le nombre est divisible par 13.

Pour vérifier :

<p>Pour $n=224185$? En effet : $22418 + (4 * 5) = 22438$ $2243 + (4 * 8) = 2275$ $227 + (4 * 5) = 247$ $24 + (4 * 7) = 52.$ 52 est un multiple de 13, donc 224185 est divisible par 13.</p>	<p>Pour $n=6355$? En effet : $635 + (4 * 5) = 655$ $65 + (4 * 5) = 85.$ 85 n'est pas un multiple de 13, donc 6355 est non divisible par 13.</p>
---	---

Travail demandé : Donner l'**algorithme** de la fonction **verif (n : entier long)** qui vérifie si un entier n est **divisible par 13** en utilisant le principe décrit précédemment.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Partie II : 12 Points

Une collection de livres est une série de 9 livres appartenant à la même maison d'édition et ayant la même couleur de couverture (**Rose** pour les petits enfants, **Verte** pour les jeunes, **Bleu** pour les moins jeunes ...) numérotés séquentiellement de 1 à 9.

Dans notre bibliothèque familiale nous disposons de quelques livres de la collection **Rose** et la collection **Verte**, tous différents.

Pour bien gérer notre bibliothèque nous nous proposons les traitements suivants :

- 1- Saisir dans un tableau **T** les codes des **N** livres existants dans la bibliothèque (**N** est compris entre 10 et 18), sachant que le code attribué à un livre commence impérativement par la lettre « **R** » pour les livres **Roses** et par la lettre « **V** » pour les livres **Verts**, suivie du numéro d'ordre du livre dans sa collection, tout en respectant les conditions déjà mentionnées.

Exemple : le code R3 est le code du livre numéro 3 de la collection Rose.

- 2- Ranger les codes des livres **Roses** à gauche du tableau **T**, et ceux des livres **Verts** à sa droite en les triant dans l'ordre croissant de leur numéro d'ordre dans la collection.
- 3- Afficher pour chacune des deux collections **Rose** et **Verte** les numéros des livres manquants dans la bibliothèque.

Exemple : pour $N = 12$

Etape 1 :

T	R2	V1	V6	R1	V2	V4	R3	R6	R7	R5	V3	R9	V7
	1	2	3	4	5	6	7	8	9	10	11	12	13

Etape 2 :

T	R1	R2	R3	R5	R6	R7	R9	V1	V2	V3	V4	V6	V7
	1	2	3	4	5	6	7	8	9	10	11	12	13

Etape 3 :

Le programme affichera :

Les livres de la collection Rose manquants dans la bibliothèque sont :

4 – 8

Les livres de la collection Verte manquants dans la bibliothèque sont :

5 – 8 – 9

Travail à faire :

- a) Analyser le problème en le décomposant en modules.
- b) Analyser chacun des modules envisagés ci-dessus.

CORRECTION

REPUBLIQUE TUNISIENNE
 MINISTERE DE L'EDUCATION
 ♦♦♦♦♦
DEVOIR DE SYNTHESE N° :3
10/05/2016

Epreuve : INFORMATIQUE
 ✕ ✕ ✕ ✕
 Durée : **1 h 30**
 ✕ ✕ ✕ ✕
 Coefficient : 1

SECTIONS : MATHÉMATIQUES, SCIENCES EXPÉRIMENTALES ET SCIENCES TECHNIQUES

Nom et prénom : Classe : 4^{ème} N° :

Partie I : 8 Points

Exercice N°1 : 4 Points

Soit le tableau **PAT** qui donne le nombre de patients ayant visités un certain cabinet médical par mois :

292	300	231	182	169	143	151	128	156	169	192	251
Janvier	Février	Mars	Avril	Mai	Juin	Juillet	Août	Septembre	Octobre	Novembre	Décembre

Questions :

1) En utilisant le type scalaire énuméré, déclarer en **Pascal** un type **Mois** contenant les mois de l'année.
Type

Mois=(Janvier, Fevrier, Mars, Avril, Mai, Juin, Juillet, Aout, Septembre, Octobre, Novembre, Decembre);

2) Soit **TAB** le type du tableau **PAT**, déclarer en **Pascal** le type **TAB**.

Type

TAB=Array[Janvier..Decembre] of Integer;

3) Compléter le tableau ci-dessous en mettant une croix (X) dans la colonne **valide** ou dans la colonne **non valide** pour chaque instruction. **Justifier votre réponse pour les instructions non valides.**

Déclarations / instructions	Valide	Non valide	Justification
WriteLn(PAT[Mai]);	X		
Var i:string; begin For i:= Janvier to Decembre do PAT[i]:=2*PAT[i];		X	Le compteur i doit être de type Mois (scalaire énuméré)
Function Max(p:TAB):Mois;	X		
Var M:Mois; begin ReadLn(M);		X	On ne peut ni lire ni écrire des valeurs de type Mois (scalaire énuméré)

Exercice N°2 : 4 Points

Un entier n ($n \geq 10$) est **divisible par 13**, si on multiplie successivement le dernier chiffre par 4 et on l'additionne au nombre privé de son dernier chiffre, le traitement s'arrête lorsque le résultat obtenu est composé de 2 chiffres, s'il est un multiple de 13, alors le nombre est divisible par 13.

Pour vérifier :

Pour $n=224185$?

En effet : $22418 + (4 * 5) = 22438$

$$2243 + (4 * 8) = 2275$$

$$227 + (4 * 5) = 247$$

$$24 + (4 * 7) = 52.$$

52 est un multiple de 13, donc 224185 est divisible par 13.

Pour $n=6355$?

En effet : $635 + (4 * 5) = 655$

$$65 + (4 * 5) = 85.$$

85 n'est pas un multiple de 13, donc 6355 est non divisible par 13.

Travail demandé : Donner l'algorithme de la fonction **verif (n : entier long)** qui vérifie si un entier n est divisible par 13 en utilisant le principe décrit précédemment.

Méthode N°1 :

```
0) DEF FN Verif(N :Entier long) :Booléen
1) Tant Que N >= 100 Faire
    Convch(N,Ch)
    Valeur(SousChaîne(Ch,1,Long(Ch)-1),X,E)
    Valeur(Ch[Long(Ch)],Y,E)
    N ← X + 4 * Y
Fin Tant Que
2) Si N Mod 13 = 0 Alors
    Verif ← Vrai
Sinon
    Verif ← Faux
FinSi
3) Fin Verif
```

Méthode N°2 :

```
0) DEF FN Verif(N :Entier long) :Booléen
1) Tant Que N >= 100 Faire
    N ← N DIV 10 + 4 * (N MOD 10)
Fin Tant Que
2) Si N Mod 13 = 0 Alors
    Verif ← Vrai
Sinon
    Verif ← Faux
FinSi
3) Fin Verif
```

Partie II : 12 Points

Analyse du P.P :

Nom : Gestion_Bibliotheque

Résultat= { Affichage des livres manquant de la collection Rose}

Proc Affichage_manquant(T,N,"R","Rose")

{Affichage des livres manquant de la collection Verte}

Proc Affichage_manquant(T,N,"V","Verte")

{Tri du tableau T}

T=Proc Tri(T,N)

{Remplissage du tableau T}

T=Proc Remplir(T,N)

{Saisir la valeur de N}

N= Proc Saisir(N)

Fin Gestion_Bibliotheque

Analyse de la procédure Saisir

DEF PROC Saisir(VAR N :Entier)

Résultat=N

N=[] Répéter

N=Donnée("Donner le nombre des livres")

Jusqu'à N Dans [10..18]

Fin Saisir

Analyse de la procédure Remplir

DEF PROC Remplir(VAR T :Tab ; N :Entier)

Résultat=T

T=[] Pour *i* de 1 à N Faire

Répéter

T[i]=Donnée("Donner le code du livre N°",i, "; ")

Jusqu'à (T[i][1] dans ["R".."V"]) ET (T[i][2] dans ["1".."9"]) ET (NON(*Fn Exist*(T[i],T,i-1)))

Fin Pour

Fin Remplir

T.D.N.T

Type
TAB=Tableau [1..18] de chaîne de Caractère

T.D.O.G

Objets	Type/Nature
N	V/Entier
T	TAB
Saisir	Procédure
Remplir	Procédure
Tri	Procédure
Affichage_manquant	Procédure
Exist	Fonction

T.D.O.L

Objets	Type/Nature
i	V/Entier

Analyse de la fonction Exist

DEF FN Exist(Ch :Chaîne de caractère ; T :Tab ; N :Entier)

Résultat=Exist

Exist=[] *Si* $i \leq N$ *Alors*

Exist ← Vrai

Sinon

Exist ← Faux

Fin Si

T=[$i \leftarrow 1$] *Tant Que* ($T[i] \neq \text{Ch}$) *ET* ($i \leq N$) *Faire*

$i \leftarrow i + 1$

Fin Tant Que

Fin Exist

T.D.O.L

Objets	Type/Nature
<i>i</i>	V/Entier

Analyse de la procédure Tri

DEF PROC Tri(VAR T :Tab ; N :Entier)

Résultat=T

T=[] *Répéter*

Echange ← Faux

Pour *i* de 1 à N-1 *Faire*

Si $T[i] > T[i+1]$ *Alors*

T = Proc *Permut*(*T*[*i*], *T*[*i*+1])

Echange ← Vrai

Fin Si

Fin Pour

$N \leftarrow N - 1$

Jusqu'à (NON(*Echange*)) OU ($N = 1$)

Fin Tri

T.D.O.L

Objets	Type/Nature
<i>i</i>	V/Entier
<i>Echange</i>	V/Booléen
<i>Permut</i>	Procédure

Analyse de la procédure Permut

DEF PROC Permut(VAR Ch1, Ch2 :Chaîne de caractère)

Résultat=Ch1, Ch2

Aux ← Ch1

Ch1 ← Ch2

Ch2 ← *Aux*

T.D.O.L

Objets	Type/Nature
<i>Aux</i>	V/Chaîne de caractère

Fin Permut

Analyse de la procédure Affichage_manquant

DEF PROC Affichage_manquant(T :Tab ; N :Entier ; C :Caractère ; Ch :Chaîne de caractère)

Résultat=Affichage_manquant

Affichage_manquant=[Ecrire("Les livres de la collection "Ch," manquants dans la bibliothèque sont :"]

Pour i de 1 à 9 Faire

ConvCh(i,Code)

Code ← C+Code

Si Non(Fn Exist(Code,T,N)) Alors

Ecrire(i, "-")

Fin Si

Fin Pour

Fin Affichage_manquant

T.D.O.L

Objets	Type/Nature
i	V/Entier
Code	V/Chaîne de caractère

