

<u>Date</u> : 04 /12/2012, à 8h <u>Durée</u> : 2 heures	<h2 style="margin: 0;">Devoir De Synthèse</h2> <h3 style="margin: 0;">n° 1</h3> <h4 style="margin: 0;">Algorithmique et Programmation</h4>	<u>Classe</u> : 4 <sup>ème</sup> SI
<u>Proposé par</u> : M <sup>r</sup> T <sup>r</sup> abelsi Karim / M <sup>me</sup> K <sup>h</sup> ichi Rim / M <sup>r</sup> B <sup>r</sup> ouk Achraf		
<u>Nom et prénom</u> : .....		

**Important :**

- L'examen est composé de 3 exercices et un problème
- Les exercices 1,2 et 3 sont à reproduire sur la feuille de l'examen qu'il faudra rendre à la fin de l'épreuve.
- Le problème est à développer sur une double feuille.
- Les pages sont numérotées de 1/4 jusqu'à 4/4

**Exercice n° 1 : (2.5 pts)**

Soit la partie déclarative suivante, écrite en algorithmique :

T.D.N.T
<b>Eleve</b> = Enregistrement - Nom_Pr : Chaîne [30] - Date_N : Chaîne [10] - Notes : tableau de 3 réels - Moy : réel <b>Fin Eleve</b>
<b>Tab</b> =Tableau de 30 Eleve
<b>FEleve</b> =Fichier d'Eleve
<b>Fnombre</b> =Fichier d'entier

Objet	Type	Rôle
E	Eleve	Enregistrement concernant un élève
FE	FEleve	Fichier contenant des élèves
T	Tab	Tableau d'élèves
Fn	Fnombre	Fichier contenant des entiers
Ft	Texte	Fichier texte
drapeau	booléen	Variable logique

Compléter par « **Oui** » si l'expression est permise et par « **Non** » sinon, justifier votre réponse.

Expression algorithmique	Oui/Non	Justification si vous répondez « Non »
T[3].Nom_Pr ← "Mohamed Ali"		
Lire (E)		
Ecrire (Position_Fichier (Fn)-1)		
Ecrire (sous_chaine(E.Date_N,3,2))		
drapeau ← Taille_Fichier(Fn)		
T[1].Notes ← 12.8		
Ecrire_nL(Ft,Moy)		

## Exercice n° 2 : (3.5 pts)

Soit l'algorithme de la fonction **inconnue** suivante :

```
0) Def Fn inconnue (T :tab ;d,f :Entier) :.....
1) pm ← d
2) Pour i de (d+1) à f Faire
    Si (T[i] > T[pm]) Alors
        pm ← i
    FinSi
  FinPour
3) inconnue ← pm
4) Fin inconnue
```

### N.B. :

**Tab** est un type utilisateur tableau pouvant contenir au maximum **50** réels

### Questions :

- 1) Déterminer le type de la fonction **inconnue**. (0.25pt)
- 2) Compléter le tableau de déclaration des objets locaux à la fonction **inconnue**, par les déclarations nécessaires : (0.5pt)

Objet	Type/Nature

- 3) Pour le tableau **T** suivant, calculer : **inconnue (T, 2,5)** en laissant la trace d'exécution. (0.5 pt)

i	1	2	3	4	5	6	7
T[i]	-2	5.2	-1.6	12	7.8	125	30

- 4) Dédire le rôle de la fonction **inconnue**. (0.5pt)

- 5) Proposer un algorithme **récurif** pour la fonction **inconnue**. (1.75 pts)

## Exercice n° 3 : (2.5 pts)

Ecrire l'algorithme d'une fonction **réursive** qui supprime tous les chiffres d'une chaîne **CH** donnée.

**Exemple** : Pour  $CH="B12ac-S1i3"$  → La fonction retournera la chaîne : "**Bac-Si**"

## Problème : (11.5 pts)

Le code **EAN13** (*European Article Numbering à 13 chiffres*) est un code à barres utilisé sur l'ensemble des produits de grande consommation. Ce code est une chaîne composée de **13** chiffres.

Un code **EAN13** est formé par:

- Un identifiant du produit **q** formé par les **12** premiers chiffres à gauche.
- La clé de contrôle **cc** formé par le dernier chiffre à droite.

Pour vérifier qu'un nombre de 13 chiffres est un code **EAN13** valide on applique le principe suivant :

1. On calcule la somme **S** des chiffres de **q** en commençant par le chiffre le plus à gauche et en multipliant les chiffres **de rang pair** par **3**. Le rang du premier chiffre le plus à gauche est **1**, celui du deuxième chiffre le plus à gauche est **2**, etc.
2. On calcule le reste **r** de la division euclidienne de **S** par **10**.
3. On calcule **p** qui est égal à **(10 - r)**.
4. Si **p** est égal à **cc** alors le code est valide.

### Travail demandé :

On se propose d'écrire un programme qui permet de remplir un fichier texte « **code.txt** » par **n** (avec  $2 \leq n \leq 30$ ) nombres à **13 chiffres** chacun, à raison d'un nombre par ligne, puis de construire un deuxième fichier de données « **res.fch** » contenant autant d'enregistrements que de nombres dans le fichier texte « **code.txt** ». Chaque enregistrement du fichier « **res.fch** » est identifié par deux champs :

- **Code** : contenant le nombre qui sera pris du fichier « **code.txt** »
- **EAN** : contient la valeur **vrai** si le nombre pris du fichier « **code.txt** » est un code **EAN13**, **faux** sinon.

A partir du fichier « **res.fch** », afficher les nombres qui sont des codes **EAN13**.

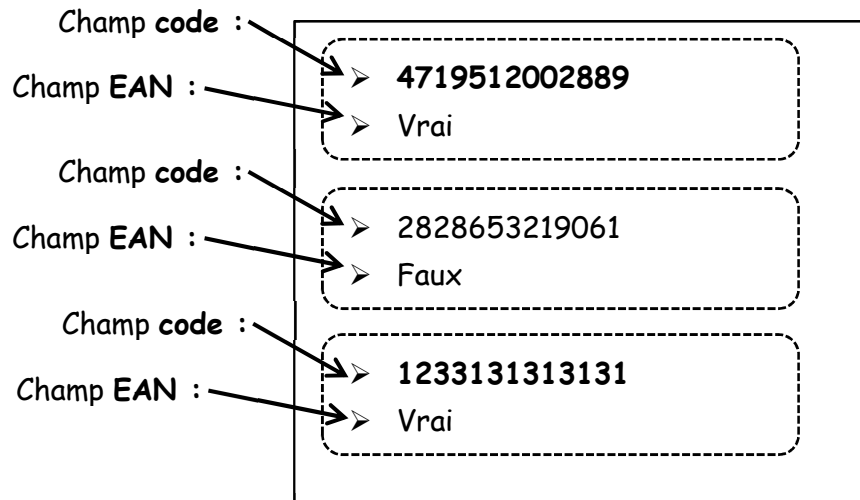
### N.B. :

- On suppose que tous les fichiers seront mis à la racine du lecteur **C**.

Exemple : Pour  $n=3$  et pour le fichier texte « **code.txt** » suivant :

4719512002889
2828653219061
1233131313131

Alors le fichier « **res.fch** » contiendra :



**En effet :**

- Appliquons la démarche précédente sur le nombre  $\overbrace{471951200288}^q$  9 ← cc
  - $cc = 9$
  - $q = 471951200288$
  - $S = 4 + (7 \cdot 3) + 1 + (9 \cdot 3) + 5 + (1 \cdot 3) + 2 + (0 \cdot 3) + 0 + (2 \cdot 3) + 8 + (8 \cdot 3) = 101$
  - Le reste de la division euclidienne de  $S$  par 10 donne  $r = 1$
  - $p = 10 - r = 9$
  - On remarque que  $p = cc$  donc le nombre **4719512002889** est un code **EAN13**.
- Appliquons la démarche précédente sur le nombre **2828653219061** :
  - $cc = 1$
  - $q = 282865321906$
  - $S = 2 + (8 \cdot 3) + 2 + (8 \cdot 3) + 6 + (5 \cdot 3) + 3 + (2 \cdot 3) + 1 + (9 \cdot 3) + 0 + (6 \cdot 3) = 128$
  - Le reste de la division euclidienne de  $S$  par 10 donne  $r = 8$
  - $p = 10 - r = 2$
  - On remarque que  $p \neq cc$  donc le nombre **2828653219061** n'est pas un code **EAN13**.
- Appliquons la démarche précédente sur le nombre **1233131313131** :
  - $cc = 1$
  - $q = 123313131313$
  - $S = 1 + (2 \cdot 3) + 3 + (3 \cdot 3) + 1 + (3 \cdot 3) + 1 + (3 \cdot 3) + 1 + (3 \cdot 3) + 1 + (3 \cdot 3) = 59$
  - Le reste de la division euclidienne de  $S$  par 10 donne  $r = 9$
  - $p = 10 - r = 1$
  - On remarque que  $p = cc$  donc le nombre **1233131313131** est un code **EAN13**.

### Questions :

- 1) Analyser ce problème en le décomposant en modules et en respectant la démarche proposée dans les énoncés et en déduire l'algorithme du programme principal
- 2) Analyser les modules envisagés