

## Les fichiers binaires

### Activité 1 : Création, remplissage d'un fichier binaire avec un nombre de lignes connu et affichage avec un nombre de lignes inconnus

Un nombre **Nb** est dit **rigolo** si la somme de ses chiffres est égale à la somme de tous les chiffres de ses facteurs premiers.

**Exemple 1 :**

Si **Nb = 690**, alors **Nb** est un nombre **rigolo**. En effet :

- La somme des chiffres de **690** est :  $6 + 9 + 0 = 15$
- Les facteurs premiers de **690** sont : 2, 3, 5 et 23
- La somme des chiffres des facteurs premiers de **690** est :  $2 + 3 + 5 + 2 + 3 = 15$

**Exemple 2 :**

Si **Nb = 120**, alors **Nb** n'est pas un nombre **rigolo**. En effet :

- La somme des chiffres de **120** est :  $1 + 2 + 0 = 3$
- Les facteurs premiers de **120** sont : 2, 2, 2, 3 et 5
- La somme des chiffres des facteurs premiers de **120** est :  $2 + 2 + 2 + 3 + 5 = 10$

Ecrire un programme qui

- Remplir un fichier **nombres.dat** par tous les nombres de 1 à 1000
- Chercher les nombres **rigolos** du fichier **nombres.dat** et les stocker dans un fichier d'enregistrements **rigolo.dat** sachant qu'un enregistrement du fichier rigolo comporte :
  - Le champ **valeur** du nombre
  - Le champ Facteur premier du nombre stocké sous la forme **f1\*f1\*.....\*fn**

### Activité2 : Fichier binaire\enregistrement\Tri par insertion

Soit l'algorithme « Tri\_insertion » suivant permettant de trier par ordre croissant un tableau **T** rempli par **N** entiers.

Procédure Tri\_insertion (@T : TAB , N : entier)

Début

Pour i de 1 à N-1 faire

**V** ← T[i]

  j ← i

  Tant que (T[j-1] > V) et (j > 0) faire

    T[j] ← T[j-1]

    j ← j-1

  Fin tant que

**T[j] ← V**

Fin pour

Fin



Essayez de comprendre le principe de la procédure tri\_insertion

On vous demande de faire le programme permettant de :

1 - Remplir un fichier F « **classe.dat** » par les informations de N élèves sachant que  $2 < N < 10$  et que chaque élève est un enregistrement présentant les données suivantes :

<i>ELEVE</i>		
<i>Champ</i>	<i>Signification</i>	<i>Type</i>
<i>id</i>	Identifiant	Entier
<i>no</i>	Nom	Chaine formée par des lettres alphabétiques
<i>pr</i>	Prénom	Chaine formée par des lettres alphabétiques
<i>mg</i>	Moyenne Générale	Réel compris entre 0 et 20

2- Trier le contenu du fichier par ordre décroissant

3- Afficher les élites de la classe (les élèves ayant une moyenne générale  $\geq 15$ ) par ordre de mérite.

NB : pour faire le tri du fichier il faut suivre les étapes suivantes :

- 1- Transférer le contenu du fichier vers un tableau d'enregistrements.
- 2- Trier le contenu du tableau T par ordre décroissant
- 3- Transférer le contenu du tableau trié vers le fichier.

Travail à faire :

- 1- Donner l'algorithme du programme principal
- 2- Donner les algorithmes des modules envisagés

## Exercice 1

On désire créer un fichier de données pour stocker la liste des notes des élèves d'une classe dans le fichier « **D:\notes.bin** »

- 1- Ecrire l'algorithme de la procédure **Remplir (@F : Fich)** pour chacune des opérations suivantes :
  - a- Ecriture d'une seule note dans le fichier des élèves.
  - b- Ecriture de N notes dans le fichier des N élèves. (N données)
  - c- Ecriture dans le fichier des élèves jusqu'à ce que l'utilisateur saisisse une note négative.
  - d- Ecriture dans le fichier des élèves jusqu'à ce que l'utilisateur demande d'arrêter la saisie.
- 2- Ecrire l'algorithme d'un module pour chacune des opérations suivantes :
  - a- Afficher la liste des notes sur écran.
  - b- Afficher la meilleure note ainsi que la moyenne des notes.
  - c- Afficher l'existence d'une donnée X dans le fichier.
  - d- Trier le fichier dans l'ordre décroissant des notes.

## Problème (Session Principale 2012)

**Louis Braille**, est l'inventeur du système d'écriture tactile à points saillants, à l'usage des personnes aveugles ou fortement malvoyantes.

En Braille standard :

- Un caractère est représenté par six points numérotés de 1 à 6 et disposés comme le montre la **Figure 1**.
- Un point peut être saillant (en relief) ou non, comme le montre la **Figure 2**.
- Le nombre et la disposition des points en relief définissent un caractère.

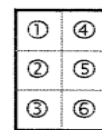


Figure 1

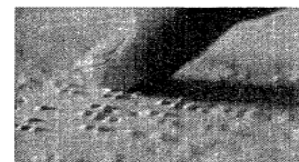




























Figure 2

Dans la suite, on s'intéressera à la représentation des 26 lettres majuscules de l'alphabet français. Le tableau suivant, donne cette représentation.

A 	B 	C 	D 	E 	F 	G 	H 
I 	J 	K 	L 	M 	N 	O 	P 
Q 	R 	S 	T 	U 	V 	W 	X 
Y 	Z 						

**N. B** Chaque point noir représente un point saillant.

Etant donné un fichier d'enregistrements intitulé "**Codes\_Braille.dat**", où chaque enregistrement est composé de deux champs :

- un champ **Lettre** contenant une lettre majuscule de l'alphabet français,
- un champ **Codage** contenant une chaîne de 6 caractères représentant l'équivalent en braille de la lettre.

En utilisant le fichier "**Codes\_Braille.dat**", on se propose de convertir le fichier texte intitulé "**Braille.txt**" contenant une représentation Braille d'un texte en son équivalent en alphabet français puis d'afficher le résultat obtenu.

Sachant que :

- Chaque ligne du fichier **"Braille.txt"** contient la représentation d'un seul mot.
- La représentation d'un mot est une concaténation de blocs de six caractères.
- Chaque bloc de six caractères représente une lettre du mot.
- Un caractère peut être un astérisque ("\*") représentant un point saillant, ou un trait d'union ("-") représentant un point non saillant.
- Les caractères "\*" et "-" sont disposés selon l'ordre des numéros des points qu'ils représentent. Par exemple, la lettre **"H"** sera représentée par le bloc de six caractères suivant :

$$\begin{array}{cccccc} * & * & \bar{\phantom{0}} & \bar{\phantom{0}} & * & \bar{\phantom{0}} \\ 1 & 2 & 3 & 4 & 5 & 6 \end{array}$$

**Exemple :**

Etant donné le contenu du fichier "**Codes\_Braille.dat**" dont une partie est représentée comme suit :

A \* \_ \_ \_ \_  
B \*\* \_ \_ \_ \_  
C \* \_ \_ \* \_ \_  
D \* \_ \_ \*\* \_ \_

Si le contenu du fichier "**Braille.txt**" est le suivant :

\* \_ \_ \* \_ \* \_ \* \_ \* \_ \_ \_ \* \_ \* \_ \* \_ \_ \* \_ \* \_ \* \_  
\* \_ \_ \* \_ \* \_ \* \_ \*  
\* \_ \_ \_ \_ \* \_ \_ \_ \_ \* \_ \* \_ \_

Le programme affichera la chaîne : **"EXAMEN DU BAC"**

En effet :

- "EXAMEN" est l'équivalent en alphabet français de la première ligne du fichier "Braille.txt".

E X A M E N  
\* \* \* \* \* \* \* \* \*

- "DU" est l'équivalent en alphabet français de la deuxième ligne du fichier "**Braille.txt**".
- "BAC" est l'équivalent en alphabet français de la troisième ligne du fichier "**Braille.txt**".

**N. B** Le candidat n'est pas appelé à remplir les deux fichiers "**Codes\_Braille.dat**" et "**Braille.txt**".

**Travail demandé :**

1. Analyser le problème en le décomposant en modules.
2. Analyser chacun des modules envisagés.