

◆
DEVOIR DE SYNTHÈSE N°2

☆☆☆☆
Année Scolaire : 2010/2011

Section : **Sciences de l'informatique**

Epreuve : **Algorithmiques et programmation**

Durée : **3 h**

Coefficient : **3**

Date : **Vendredi, 13 mai 2011**

Le sujet comporte 3 pages

Exercice 1 (3 points)

Soit la procédure Pascal suivante permettant de trier un fichier F contenant des entiers :

```
Procédure Tri (Var F : FI) ; {avec FI : file of integer}
Var
  I, X, Y, N : Integer;
  C : Boolean;
Begin
  Reset (F);
  N := Filesize (F);
  Repeat
    C := False;
    For I := 1 to N-1 Do
      Begin
        Seek (F, I-1);
        Read (F, X);
        Read (F, Y);
        If (X > Y) Then
          Begin
            Seek (F, I-1);
            Write (F, Y);
            Write (F, X);
            C := True;
          End;
        End;
      End;
    N := N-1;
  Until C = False;
End;
```

Questions :

- 1- Quelle est la méthode de tri utilisée dans ce module ? Donner brièvement le principe de cette méthode.
- 2- Quel est l'ordre de tri utilisé (croissant ou décroissant) ?
- 3- Donner le rôle de chacune des modules prédéfinis suivants :

Fonction	Rôle
SEEK	
FILESIZE	
RESET	

Exercice 2 (3 points)

Soit la suite U définie par :

$$\begin{cases} U_1 = 1 \\ U_2 = 2 \\ U_i = U_{i-1} + (a * U_{i-2}) \text{ Pour } i > 2 \text{ et } a \text{ est un entier strictement positif donné.} \end{cases}$$

Questions :

- 1- Quel est l'ordre de récurrence de cette suite ? Justifier votre réponse.
- 2- Ecrire un algorithme d'une fonction permettant de vérifier si un entier n donné est un terme de la suite U .

Exercice 3 (4 points)

On appelle « *matrice triangulaire supérieure* » une matrice carrée M ($n \times n$) pour laquelle tout $M[L,C]=0$ si $L < C$ (Tous les éléments situés au-dessous de la diagonale sont égaux à zéro).

On appelle « *matrice triangulaire inférieure* » une matrice carrée M ($n \times n$) pour laquelle tout $M[L,C]=0$ si $L > C$ (Tous les éléments situés au-dessus de la diagonale sont égaux à zéro).

Ecrire un algorithme d'un module qui vérifie si une matrice carrée M ($n \times n$) donnée est : triangulaire supérieure, triangulaire inférieure ou autre.

Exemples : Soit les deux matrices carrées (3×3) $M1$ et $M2$:

M1	1	2	3
1	-	-	-
2	0	-	-
3	0	0	-

M2	1	2	3
1	-	0	0
2	-	-	0
3	-	-	-

Alors $M1$ est une matrice triangulaire supérieure

$M2$ est une matrice triangulaire inférieure

Problème (10 points)

Soit à remplir au hasard une matrice carrée d'ordre n ($3 < n < 23$) par les entiers **0** ou **1**.

NB : La fonction *Random* (b) avec b un entier naturel, permet de générer un entier aléatoire appartenant à l'intervalle $[0, b-1]$.

À partir de cette matrice, on désire créer un fichier d'enregistrements **FH** stocké physiquement sous "c:\Hexa.dat". Chaque enregistrement de ce fichier renferme les deux champs suivants :

- **Hex** : représente la valeur obtenue suite à la conversion en hexadécimal de chaque ligne de la matrice. La conversion en hexadécimal se fait directement et sans passer par la base 10 (regroupement par 4 de droite à gauche).
- **nb** : représente le nombre de lettres existant dans la valeur hexadécimale (Hex).

Il s'agit d'afficher enfin le contenu du fichier **FH**.

Exemple : Pour $n = 9$

M	1	2	3	4	5	6	7	8	9	FH	
1	1	1	1	0	0	1	1	1	0	1CE	2
2	1	0	0	0	1	1	1	1	1	11F	1
3	1	1	1	1	1	1	1	1	1	1FF	2
4	0	0	1	1	0	0	0	1	0	62	0
5	1	1	0	0	1	1	0	0	1	199	0
6	0	1	1	1	0	1	1	1	1	EF	2
7	0	0	1	0	0	0	0	1	0	42	0
8	1	1	1	1	1	1	0	1	0	1FA	2
9	1	1	1	0	0	0	1	1	1	1C7	1

La première ligne de la matrice **M** contient **111001110**. Son équivalent en hexadécimal est **1CE**. Cette valeur hexadécimale contient **2** lettres (C et E).

Questions :

- 1- Analyser le problème en le décomposant en modules et en déduire l'algorithme du programme principal.
- 2- Analyser chacun des modules envisagés précédemment et en déduire l'algorithme d'un module de votre choix.

Bon travail 