

CONTRÔLE FINAL (DURÉE : 1H 30 MN)**QUESTIONS DE COURS : (04 pts)**

- 1- Quelle est la stratégie d'ajout et de suppression des éléments dans : **liste, pile et file** (1.5)
- 2- Quelle est la différence entre un **tableau** et un **enregistrement**.(1)
- 3- On considère la suite définie par :

$$\begin{cases} U_0 = 2 \\ U_n = 3 * U_{n-1} - 1 \end{cases}$$

Ecrire une **fonction récursive** qui prend comme paramètre **n** un entier naturel et calcule le terme U_n de la suite. (1.5)

EXERCICE N°1: (06 pts)

Ecrire un algorithme qui va calculer et afficher la matrice **At** qui est la transposé d'une matrice d'entiers **A (N, M)**, (**N et M max 50**)

Exemple :

A	5	-2	6	15
	6	8	4	10
	8	7	98	-8

→

At	5	6	8
	-2	8	7
	6	4	98
	15	10	-8

EXERCICE N°2: (10 pts)

Soit **T1** un tableau d'entiers (taille max 40).

- a) Ecrire une fonction **Existe** (3 paramètres) de type booléen qui retourne si un élément existe ou non dans un tableau.
 - b) Ecrire une procédure **LireTab** (2 paramètres) qui permet de lire des éléments du tableau sans répétition. **NB . Il faut utiliser la fonction Existe pour ne pas lire un élément déjà existant.**
 - c) Ecrire une procédure **Inverse** (2 paramètres) qui permet d'inverser les éléments d'un tableau.
 - d) Ecrire une procédure **Affiche** (2 paramètres) qui permet d'afficher les éléments d'un tableau.
- En utilisant les sous-programmes précédents écrire un algorithme qui va lire un tableau (d'éléments différents) puis affiche son tableau inverse.

Exemple : T1 (après remplissage):

-6	5	-1	0	8	49	17	-8	13	25
----	---	----	---	---	----	----	----	----	----

Résultat : T1 (après inversion) :

25	13	-8	17	49	8	0	-1	5	-6
----	----	----	----	----	---	---	----	---	----

Bon courage