

## Les fichiers textes

Pour chaque problème, écrire l'algorithme du **programme principal** en le décomposant en modules, puis écrire les algorithmes des modules (sous programmes) envisagés

### Activité 1 : Création, remplissage et affichage du fichier texte avec un nombre de lignes connu

Le responsable du « Parc du Belvédère » de Tunis veut sauvegarder la liste des animaux du parc dans un fichier, pour cela il demande votre aide pour lui faire un programme nommé **Zoo** qui permet de :

- • Créer un fichier nommé physiquement « **animaux.txt** » et logiquement **F**
- • Sauvegarder dans le fichier **F** les noms de N animal sachant que  $2 < N < 30$  à raison d'un animal par ligne.
- • Afficher à l'écran la liste des animaux. (Le nom d'un animal par ligne)

### Activité2 : Remplissage, lecture d'un fichier texte sans connaitre son nombre de lignes

Créer un programme qui effectue les tâches suivantes :

- Création et saisie d'un fichier texte enregistré sur **D:** sous le nom "**Original.txt**".
- Copier le contenu de ce fichier dans "**Copie.txt**" avec numérotation des lignes.

### Activité 3 : Ajout, Modification, Suppression d'une ligne dans un fichier texte

Ecrire les modules qui permettent de créer un fichier texte nommé "**opération.txt**" puis permettent à l'utilisateur les opérations suivantes au choix :

- 1- L'ajout d'une ligne
  - 2- L'affichage des lignes numérotées
  - 3- La suppression d'une ligne
  - 4- La modification d'une ligne
- 1- Algorithme de la procédure saisie :

### Exercice1 : Fichier texte=>Tableau d'enregistrements

Ecrire un programme qui remplit un fichier par des phrases (une phrase est une chaîne de caractères qui contient des caractères alphabétiques et au moins un espace, commence par une lettre majuscule et se termine par un point) puis remplit et affiche un tableau d'enregistrements qui contient :

- Le numéro de la ligne de la phrase
- Le nombre de mots de cette phrase

**NB :** après chaque saisie, l'ordinateur demande à l'utilisateur de continuer la saisie d'une nouvelle phrase par une question (Voulez-vous continuer ? (**O/N**))

## Exercice 2 : Fichier texte=>Matrice

Ecrire un programme qui remplit une matrice de  $n \times n$  ( $4 \leq n < 10$ ) entiers aléatoires strictement positif et  $< 100$ , puis crée un fichier texte qui contient la liste des nombres divisibles par 3 de la matrice ligne par ligne (les entiers d'une même ligne sont séparés par le caractère #)

## Problème 1 : Fichier texte\Cryptage\Matrice BacControle2011

Dans le cadre de la réalisation d'un projet scientifique, deux chercheurs résidant dans deux pays différents, s'échangent des données sur Internet. Pour assurer la confidentialité de leur projet, les messages échangés sont cryptés.

Le message initial se compose au maximum de **50** lignes et chaque ligne contient **50** caractères au maximum. Ce message est stocké dans un fichier texte intitulé « **TxtInit.txt** » sur la racine du lecteur **C**. En supposant que ce texte est déjà saisi, on se propose de créer un deuxième fichier texte intitulé « **TxtCryp.txt** » contenant le message qui sera crypté de la manière suivante :

### 1<sup>ère</sup> étape :

Remplir une matrice d'entiers **MC** par les codes ASCII (en base 10) des caractères du message initial. Chaque ligne de la matrice contiendra les codes ASCII des caractères d'une ligne du fichier « **TxtInit.txt** ». Le nombre de lignes de la matrice **MC** est égal au nombre de lignes du fichier « **TxtInit.txt** » et le nombre de colonnes de la matrice **MC** est égal à la longueur (**long\_max**) de la plus longue ligne du fichier.

Pour les lignes ayant une longueur inférieure à **long\_max**, on complète les cases restées vides par le code ASCII du caractère **espace** (qui est égal à 32).

### 2<sup>ème</sup> étape :

Convertir, dans la même matrice, tous les éléments de **MC** en base 8.

### 3<sup>ème</sup> étape :

Stocker dans « **TxtCryp.txt** », les colonnes de la matrice **MC**, de la colonne la plus à gauche à la colonne la plus à droite. Chaque colonne de la matrice **MC** occupera une ligne et les éléments de cette ligne seront séparés par le caractère **espace**.

### Exemple :

Si le fichier « **TxtInit.txt** » contient le texte suivant,

Bonjour
Test1
Merci

alors les étapes de détermination du fichier « **TxtCryp.txt** » seront les suivantes :

**1<sup>ère</sup> étape :** A partir du fichier « **TxtInit.txt** » précédent, on obtient la matrice **MC** suivante :

	1	2	3	4	5	6	7
1	66	111	110	106	110	117	114
2	84	101	115	116	49	32	32
3	77	101	114	99	105	32	32

Où **66** est le code ASCII de « **B** », **111** celui de « **o** », ... et **32** le code ASCII du caractère **espace**.

**2<sup>ème</sup> étape :** Après la conversion en base 8, on obtient la matrice **MC** suivante :

	1	2	3	4	5	6	7
1	102	157	156	152	156	165	162
2	124	145	163	164	61	40	40
3	115	145	162	143	151	40	40

**162** est la conversion en base 8 de **(114)<sub>10</sub>**

3<sup>ème</sup> étape : A partir de la matrice MC précédente, on obtient le fichier « **TxtCryp.txt** » suivant :

102	124	115
157	145	145
156	163	162
152	164	143
156	61	151
165	40	40
162	40	40

La ligne n°4 du fichier « **TxtCryp.txt** » correspond à la colonne n°4 de la matrice MC.

## Travail demandé

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

## Problème 2 : Bac Pratique 2009

Soit **p** et **q** deux entiers naturels tels que  $10 < p < q < 100000$ . On se propose d'enregistrer dans un fichier texte **decomp.txt** placé sur la racine du lecteur C la décomposition en facteurs premiers de tous les entiers compris entre p et q. la première ligne du fichier **decomp.txt** contiendra le naturel p suivi d'un espace suivie du naturel q. chacune des autres lignes contiendra l'entier à décomposer suivi du signe égal suivi de sa décomposition

### Exemple

Si **p=34** et **q=36** le fichier **decomp.txt** contiendra les informations suivantes :

```
34 36
34=2*17
35=5*7
36=2*2*3*3
```

Ecrire un programme qui permet de remplir le fichier **decomp.txt** et l'afficher