

REPUBLIQUE TUNISIENNE
 MINISTERE DE L'EDUCATION
EXAMEN DU BACCALAUREAT
SESSION DE JUIN 2011

**SESSION
 DE CONTRÔLE**

SECTION : Sciences de l'Informatique
EPREUVE THEORIQUE : Algorithmique et programmation DUREE : 3h COEFFICIENT : 2.25

Section : N° d'inscription : Série :
 Nom et prénom :
 Date et lieu de naissance :

Signatures des
 surveillants

Le sujet comporte 3 pages numérotées de 1/3 à 3/3
Les réponses à l'exercice n°1 doivent être rédigées sur cette même page 1/3
qui doit être remise avec la feuille de copie

Première partie : (10 points)

Exercice n°1 : (3 points)

Soient les algorithmes suivants qui permettent de réaliser des traitements sur des chiffres d'affaires (CA), saisis par l'utilisateur.

Version 1	Version 2
0) DEBUT Version1 1) COMPTEUR ← X SOMME ← 0 Tant Que COMPTEUR < 4 Faire Ecrire("Saisir le chiffre d'affaires : ") Lire(CA) COMPTEUR ← COMPTEUR +1 SOMME ← SOMME + CA Fin Tant Que 2) Ecrire ("Somme des chiffres d'affaires : ", SOMME) 3) Fin Version1	0) DEBUT Version2 1) SOMME ← 0 2) Pour COMPTEUR de 0 à 3 Faire Ecrire("Saisir le chiffre d'affaires : ") Lire(CA) SOMME ← SOMME + CA Fin Pour 3) Ecrire ("Somme des chiffres d'affaires : ", SOMME) 4) Fin Version2

Pour chacune des questions suivantes, valider les propositions par **V** si la réponse est correcte ou par **F** dans le cas contraire.

1- **Dans la version n°1**, quelle(s) est (sont) la (les) valeur(s) de la variable **X** qui permet(tent) l'exécution de la boucle Tant Que?

- 0 3 4 5

2- **Dans la version n°1**, quelle(s) est (sont) la(les) valeur(s) de la variable **X** qui permet(tent) de saisir exactement **10** chiffres d'affaires (CA) ?

- 6 -4 6 10

3- **Dans la version n°2**, que contiendra la variable **SOMME** à la fin de l'exécution de cet algorithme si l'utilisateur saisit dans l'ordre, les valeurs **0, 10, 20** puis **30** ?

- 0 30 40 60

Exercice n°2 : (3,5 points)

Soit la suite U définie par :

$$\begin{cases} U_0 = 1 + \frac{1}{m} \text{ (avec } m, \text{ un entier strictement positif)} \\ U_n = 1 + \frac{1}{U_{n-1}} \text{ pour tout entier naturel } n \geq 1 \end{cases}$$

- 1) Ecrire l'algorithme d'une fonction intitulée **Calc_Suite** qui permet de calculer le $n^{\text{ième}}$ terme de cette suite pour un entier m . (m et n sont deux entiers saisis dans le programme appelant)
- 2) Quel est l'ordre de récurrence de cette fonction ? Justifier la réponse.

Exercice n°3 : (3,5 points)

Si ℓ et c désignent respectivement une ligne et une colonne d'une matrice d'entiers notée **Mat**, de dimension $M \times N$, on dit qu'une ligne ℓ est symétrique, si :

$$\text{Pour tout élément de cette ligne : } Mat[\ell, c] = Mat[\ell, N - c + 1]$$

Ecrire l'analyse d'un module intitulé **Nb_Lig_Sym** permettant de déterminer le nombre de lignes symétriques dans la matrice **Mat**.

Deuxième partie : (10 points)

Dans le cadre de la réalisation d'un projet scientifique, deux chercheurs résidants 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^{ère} 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^{ème} étape :

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

3^{ème} 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^{ère} 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^{ème} 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 $(114)_{10}$

3^{ème} 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. 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 les algorithmes correspondants.