RÉPUBLIQUE TUNISIENNE MINISTÈRE DE L'ÉDUCATION ***** EXAMEN DU BACCALAURÉAT	Épreuve : **ALGORITHMIQUE ET PROGRAMMATION**	
	Section : **Sciences de l'informatique**	
	Durée : **3H**	Coefficient : **2.25**
SESSION 2016	**Session principale**	

Section : ..... N° d'inscription : ..... Série : .....

Nom et prénom : .....

Date et lieu de naissance : .....

Signatures des surveillants

.....

.....

Le sujet comporte 5 pages numérotées de 1/5 à 5/5.
Les réponses à l'exercice1 doivent être rédigées sur les pages 1/5 et 2/5 qui doivent être remises avec la copie.

Exercice 1: (3,25 points)

Soient les déclarations suivantes :

Tableau de déclaration des nouveaux types

Type
Date = Enregistrement jour : 1..31 mois : 1..12 annee : 2003..2010 **fin Date** **Eleve = Enregistrement** nom : chaîne[15] moyenne : réel dat_nais : **Date** **fin Eleve** **Feleve = fichier d'Eleve**

Tableau de déclaration des objets

Objet	Type/Nature
F_El	Feleve
E	Eleve

Travail à faire :

1- On se propose d'ajouter dans le fichier **F_El** la liste des élèves présentée dans le tableau ci-dessous.

Compléter la deuxième colonne du tableau par :

- **V** dans le cas où les données relatives à l'élève ne présentent aucune anomalie.
- **F** dans le cas contraire, tout en justifiant la réponse dans la troisième colonne.

Elève	V/F	Justification
"Mohamed", 17, "12/11/2000"		
"Kefi", 14.5, "15/02/2003"		
"Ali", 16, "16/13/2009"		

2- Remplir le tableau suivant par les séquences d'instructions algorithmiques, permettant de réaliser les traitements indiqués dans la première colonne, sachant que le fichier **F_El** est initialement ouvert et son pointeur est positionné sur le premier enregistrement :

Traitement	Séquences d'instructions
Afficher le premier enregistrement du fichier **F_El**.	
Ajouter l'élève ("Tounsi", 14.5, "15/02/2004") à la fin du fichier **F_El**.	
Ajouter un point (1) à la moyenne du deuxième élève du fichier **F_El**.	

Élève
Evolution

RÉPUBLIQUE TUNISIENNE MINISTÈRE DE L'ÉDUCATION ***** EXAMEN DU BACCALAURÉAT	Épreuve : **ALGORITHMIQUE ET PROGRAMMATION**	
	Section : **Sciences de l'informatique**	
	Durée : **3H**	Coefficient : **2.25**
SESSION 2016	**Session principale**	

Exercice 2 : (3,75 points)

Soit **M** une matrice carrée d'ordre **N** (avec $4 \leq N \leq 10$), représentant une grille de jeu dans laquelle deux joueurs marquent des cases, à tour de rôle. Chaque case de cette matrice peut contenir soit 0, soit 1, soit 2.

- 0 : indique que cette case n'a pas été marquée.
- 1 : indique que cette case est marquée par le premier joueur.
- 2 : indique que cette case est marquée par le deuxième joueur.

Le jeu s'arrête :

- lorsqu'un joueur marque quatre (4) cases consécutives dans le sens horizontal ou dans le sens vertical. Ce joueur sera déclaré gagnant.
- lorsque toutes les cases sont marquées sans qu'aucun joueur n'arrive à assembler quatre (4) cases consécutives dans le sens horizontal ou dans le sens vertical. Dans ce cas, la partie est considérée nulle.

Élève Evolution

Exemple : Pour la matrice **M** d'ordre 6 suivante :

0	0	1	0	0	0
0	0	2	0	0	0
0	1	2	1	0	0
0	2	2	1	0	0
1	2	2	2	1	0
1	2	1	1	2	0

Le 2ème joueur est gagnant car il a marqué quatre cases consécutives dans le sens vertical.

Travail à faire :

Ecrire un algorithme d'un module intitulé "**Gagnant**" qui, à partir d'une matrice **M** d'ordre **N** et représentant une grille de jeu marquée par deux joueurs, affiche le résultat du jeu (premier joueur gagnant ou deuxième joueur gagnant ou partie nulle).

NB : **M** et **N** sont déjà saisis au niveau du programme appelant.

Exercice 3 : (4,5 points)

Un RIB est un code d'identification bancaire délivré par une banque à un titulaire de compte bancaire. Il est composé de 4 champs :

- **CB** (code de la banque) : 2 chiffres
- **CA** (code de l'agence) : 3 chiffres
- **NC** (numéro du compte) : 13 chiffres
- **CR** (clé RIB) : 2 chiffres

XX	XXX	XXXXXXXXXXXXX	XX
CB	**CA**	**NC**	**CR**

Pour calculer la clé RIB, on procède comme suit :

- Multiplier **N** par 100, sachant que **N** est un nombre de 18 chiffres obtenu suite à la concaténation des chiffres des champs **CB**, **CA** et **NC**.
- Calculer le reste de la division entière du nombre ainsi obtenu par 97.
- Soustraire de 97, le reste obtenu dans l'étape précédente. Le résultat de cette soustraction représente la clé de contrôle dite clé RIB, qui ne peut prendre qu'une valeur entre 01 et 97.

Un RIB est valide lorsque la clé figurant dans le RIB est égale à celle calculée en utilisant la méthode de calcul décrite ci-dessus.

Exemple :

Pour le RIB suivant :

Élève Evolution

10	407	02400675324 81	20

- N est égal à 104070240067532481
  N * 100 est égal à 10407024006753248100
- Le reste de la division entière de N par 97 est égal à 77
- 97 – 77 = 20 qui représente la clé de contrôle dite clé RIB.
  D'où, le RIB est valide puisque la clé de contrôle calculée est égale à la clé figurant dans le RIB.

Travail à faire :

Ecrire une analyse d'un module intitulé "**TRIB**" qui, à partir d'un fichier "**RIB.txt**" contenant des codes RIB, à raison d'un code par ligne, permet de :

- Remplir un fichier "**RIB_valide.txt**" par les RIB valides du fichier "**RIB.txt**".
- Trier le fichier "**RIB_valide.txt**" selon l'ordre croissant du code de la banque.

NB :

- Le candidat n'est pas appelé à remplir le fichier "**RIB.txt**".
- Le fichier "**RIB_valide.txt**" sera enregistré sur la racine du disque C.
- Le candidat peut utiliser une fonction intitulée **Mod97(CH)** ayant comme paramètre une chaîne de caractères **CH** représentant une valeur numérique très grande **N** et qui permet de retourner le reste de la division entière de **N** par **97**. Le candidat n'est pas appelé à développer cette fonction.

Exercice 4 : (8,5 points)

Le **codage de Fibonacci** est un code binaire, utilisant des termes de la suite de Fibonacci et servant essentiellement dans la compression de données.

Pour déterminer le code de **Fibonacci** d'un entier **K** strictement positif, on suit les étapes suivantes :

Etape 1 : Déterminer la liste des termes de la suite de Fibonacci inférieurs ou égaux à **K**, sachant que la suite de Fibonacci **U** est définie comme suit :

$$\begin{cases} U_1 = 1, U_2 = 1 \\ U_n = U_{n-1} + U_{n-2} \quad \text{pour } n > 2 \end{cases}$$

Etape 2 : Décomposer l'entier **K** en une somme des termes de la suite de Fibonacci déjà calculés dans l'étape précédente, tout en commençant par utiliser le plus grand terme inférieur à **K** et sans prendre en considération le premier terme de la suite (U_1).

Etape 3 : Former un code binaire à partir de la liste des termes calculée dans l'étape 1 et sans prendre en considération le premier terme de la suite (U_1) : en concaténant le caractère "1" dans le cas où le terme a été utilisé dans la somme calculée au niveau de l'étape 2 et en concaténant le caractère "0" dans le cas contraire.

Etape 4 : Ajouter à la fin du code obtenu précédemment le caractère "1" pour obtenir le code de Fibonacci.

Élève
Evolution

On se propose d'écrire un programme qui permet de saisir un entier **K** strictement positif et d'afficher le code de Fibonacci correspondant.

Exemple : Pour **K** = 50,

Etape 1 : La liste des termes de la suite de Fibonacci inférieurs ou égaux à 50 et sans prendre en considération le premier terme de la suite (U_1) est : 1, 2, 3, 5, 8, 13, 21 et 34.

Etape 2 : La décomposition de **K** sera comme suit : 50 = 34 + 13 + 3

Etape 3 : Etant donnée la liste des termes obtenue dans l'étape 1 et sans prendre en considération le premier terme de la suite (U_1) : 1, 2, 3, 5, 8, 13, 21 et 34
En concaténant le caractère "0" pour les termes 1, 2, 5, 8 et 21 qui n'ont pas été utilisés dans la somme et en concaténant le caractère "1" pour les termes 3, 13 et 34 qui ont été utilisés dans la somme, le code binaire formé est : 00100101

Etape 4 : En ajoutant à la fin du code binaire obtenu précédemment le caractère "1", le code de Fibonacci est : **001001011**

Travail à faire :

1- Analyser le problème en le décomposant en modules.

2- Analyser chacun des modules envisagés.