

Exercice 1: (3 + 3.5 = 6.5 points)

1-

Elève	V/F	Justification	Nombre de points
"Mohamed",17,"12/11/2000"	F	Valeur incompatible avec le type Date.	1
"Kefi", 14.5, "15/02/2003"	F	Valeur incompatible avec le type Date.	1
"Ali", 16, "16/13/2009"	F	Valeur incompatible avec le type Date.	1

Elève	V/F	Justification	Nombre de points
"Mohamed",17,"12/11/2000"	F	L'année doit être comprise entre 2003 et 2010.	1
"Kefi", 14.5, "15/02/2003"	V		1
"Ali", 16, "16/13/2009"	F	Le mois doit être compris entre 1 et 12	1

2-

Traitement	Séquences d'instructions	Nombre de points
Afficher le premier enregistrement du fichier F_EL.	Lire (F_El, el)	0.25
	Ecrire(el.nom,el.moyenne,el.date_nais.jour,"/",el.date_nais.mois,"/",el.date_nais.annee)	0.25
Ajouter l'élève ("Tounsi", 14.5,"15/02/2004") à la fin du fichier F_EL.	Pointer (F_El,Taille_fichier(F_El))	0.25
	e.nom ← "Tounsi"	0.25
	e.moyenne ← 14.5	0.25
	e.date_nais.jour ← 15	0.25
	e.date_nais.mois ← 2	0.25
	e.date_nais.annee ← 2004	0.25
	Ecrire (F_El, e)	0.25
Ajouter un point à la moyenne du deuxième élève du fichier F_EL.	Pointer (F_El, 1)	0.25
	Lire (F_El,e)	0.25
	e.moyenne ← e.moyenne + 1	0.25
	Pointer (F_El, 1)	0.25
	Ecrire (F_El, e)	0.25

Exercice 2 : (7.5 points)

0) **DEF PROC Gagnant (M:Tab ; N:Entier)**

- 1) Si FN trouve(M,N)=2 Alors Ecrire("Joueur 2 gagnant")
Sinon Si FN trouve(M,N)=1 Alors Ecrire("Joueur 1 gagnant")
Sinon Ecrire("Partie nulle")
FinSi
- 2) Fin PROC Gagnant

0) **DEF FN trouve (M:Tab ; N:Entier): Entier**

- 1) $T \leftarrow 0$
SI FN Horiz_verti(M,N,1) $\neq 0$ Alors $T \leftarrow$ FN Horiz_verti(M,N,1)
FinSi
SI FN Horiz_verti(M,N,2) $\neq 0$ Alors $T \leftarrow$ FN Horiz_verti(M,N,2)
FinSi
- 2) Trouve $\leftarrow T$
- 3) Fin FN Trouve

0) **DEF FN Horiz_verti (M:Tab; N,joueur: Entier): Entier**

- 1) $J_G \leftarrow 0$
 $i \leftarrow 0$
Répéter
 $i \leftarrow i+1$
 $j \leftarrow 0$
 Répéter
 $j \leftarrow j+1$
 Si (M[i,j]=joueur) ET (M[i,j+1]=joueur) ET (M[i,j+2]=joueur) ET (M[i,j+3]=joueur)
 Alors $J_G \leftarrow$ joueur
 FinSi
 SI (M[j,i]=joueur) ET (M[j+1,i]=joueur) ET (M[j+2,i]=joueur) ET (M[j+3,i]=joueur)
 Alors $J_G \leftarrow$ joueur
 FinSi
Jusqu'à (J_G=joueur) ou (j=N-3)
Jusqu'à (J_G=joueur) ou (i=N)
- 2) Horiz_verti $\leftarrow J_G$
- 3) Fin FN Horiz_verti

Barème :

Traitement	Nombre de points
Test + affichage du résultat	$(0.25+0.25)*3$
Vérification horizontale (parcours + test)	1+1
Vérification verticale (parcours + test)	1+1
Retour du résultat de la vérification	0.75+0.75
Entêtes des modules	0.5

Exercice 3 : (9 points)

DEF PROC RIB (Var f, f_v : Texte)

Résultat = f_v

f_v = PROC Remplir_RIB_valide(f, f_v)

PROC Tri(f_v)

Fin PROC RIB

DEF PROC Remplir_RIB_valide (Var f, f_v : texte)

Résultat = f_v

f_v = [Associer (f_v, "c:\RIB_valide.txt"),
Recréer (f_v), Ouvrir (f), Ouvrir (f_v)]

Tantque Non Fin_fichier (f) Faire

Lire_nl (f, R)

cle ← Sous_chaîne(R, 19, 2)

valeur (cle, cl, e);

code ← Sous_chaîne(R, 1, 18) + "00"

cr ← 97 – FN Mod97(code)

SI cr = cl Alors Ecrire_nl (f_v, R)

FinSI

FinTantque

Fermer (f)

Fermer (f_v)

Fin PROC Remplir_RIB_valide

Tableau de déclaration des objets locaux

Objet	Type / Nature
R, cle, code	Chaîne de caractères
Cl, e, n, cr	Entier
Mod97	Fonction

DEF PROC Tri (var f_v: texte)

Résultat = f_v

F_v = [Ouvrir (f_v), N ← 0]

{transfert dans un tableau}

Tantque Non Fin_fichier (f_v) Faire

N ← N+1

Lire_nl (f_v, T[N])

Fin Tantque

Pour i de 1 à N-1 Faire

ind ← 1

cbi ← Sous_chaîne(t[i], 1, 2)

Pour j de 2 à N Faire {tri du tableau}

cbj ← Sous_chaîne (t[j], 1, 2)

SI cbi > cbj Alors ind ← j

FinSi

FinPour

temp ← t[i]

t[i] ← t[ind]

t[ind] ← temp

FinPour

Ouvrir (f_v) {transfert dans le fichier}

Pour i de 1 à N Faire

Ecrire_nl (f_v, t[i])

Finpour

Fermer (f_v)

Fin PROC Tri

Tableau de déclaration des objets locaux

Objet	Type / Nature
cbi, cbj, temp	Chaîne de caractères
N, i, ind, j	Entier
T	Tableau de 100 chaîne [20]

Barème :

Traitement	Nombre de points
Remplissage du fichier "RIB_valide.txt": - Association + Création + Ouverture + Fermeture des fichiers - Parcours du fichier "RIB.txt" + Lecture d'une ligne - Calcul de la clé - Test + Ecriture	0.25*5 1 + 0.5 1 0.5+0.5
Tri du fichier "RIB_Valide.txt": - Transfert dans un tableau - Tri du tableau - Transfert des éléments du tableau trié dans le fichier	1 1 1
Entêtes des modules	0.5
TDOL	0.75

Exercice 4 : (17 points)

1) Analyse du programme principal :

Nom : Codage_Fibo

Résultat = PROC Affiche_code(M,n)

(M,n) = PROC Terme_fib(k,n,M)

PROC Coeff_terme(k,n,M)

K = PROC Saisie(k)

Fin Codage_Fibo

Tableau de déclaration des nouveaux types

Type
Tab = Matrice de 2 lignes et de 100 colonnes d'entiers

Tableau de déclaration des objets globaux

Objet	Type / Nature
M	Tab
K,n	Entier
Affiche_code	Procédure
Terme_Fibo	Procédure
Coeff_terme	Procédure
Saisie	Procédure

2) Analyse des modules envisagés

DEF PROC affiche_code(M:Tab;n:entier)

Résultat = Ecrire(ch)

ch = [ch ← ""]

Pour i de 2 à n Faire

Convch(M[2,i],c)

ch ← ch + c

Fin Pour

ch ← ch + "1"

Fin PROC affiche_code

Tableau de déclaration des objets locaux

Objet	Type / Nature
Ch	Chaîne de caractères
i	Entier
c	Caractère

DEF PROC terme_fib(k:entier ; var n:entier; var M:tab)

Résultat = M,n

(M,n) = [M[1,1] ← 1, M[1,2] ← 1, n ← 2]

Tantque M[1,n] + M[1,n-1] ≤ k Faire

n ← n + 1

M[1,n] ← M[1,n-1] + M[1,n-2]

Fin Tantque

Fin PROC terme_fib

DEF PROC coeff_terme(k,n:entier ; var M:tab)

Résultat = M

M = [s ← 0]

Si M[1,n] = k alors n ← n-1

Finsi

Pour i de n à 2 pas -1 Faire

SI (s + M[1,i]) ≤ k Alors

s ← s + M[1,i]

M[2,i] ← 1

Sinon M[2,i] ← 0

FinSI

FinPour

Fin PROC coeff_terme

Tableau de déclaration des objets locaux

Objet	Type / Nature
i, s	Entier

DEF PROC Saisie(var k:entier)

Résultat = k

k = []

Répéter

k = donnée

Jusqu'à k > 0

Fin PROC Saisie

Barème :

Traitement	Nombre de points
Décomposition	1
Appels + cohérence des paramètres	0.5 + 0.5
Saisie de k avec respect des contraintes	1.5 = 0.5 + 1
Détermination des termes de Fibonacci - Initialisation - Parcours - Calcul du nouveau terme - Incrémentation du compteur	0.75 1 1 0.25
Décomposition de K en une somme des termes de la suite de Fibonacci - Initialisation - Parcours - Test - Calcul de somme	0.5 1.5 1 1
Détermination du code de Fibonacci + Affichage - Initialisation - Parcours - Test (0 ou 1) - Concaténation - Ajout du caractère "1" - Affichage	0.5 1 0.5 0.5 0.5 1
TDNT + TDOG + TDOL	0.5 + 1 + 1