

Corrigé : Algorithmique et Programmation
Section : Sciences de l'informatique
Session de contrôle 2017

Exercice 1 : (3 points)

Question n°1 :

P	Id=123	Id=113	Id=125	Id=115	Id=363	Id=263	Id=430	Id=380	Id=455	Id=663
	Age=57	Age=57	Age=22	Age=30	Age=35	Age=18	Age=33	Age=55	Age=23	Age=19
	Genre=M	Genre=F	Genre=M	Genre=F	Genre=M	Genre=F	Genre=M	Genre=F	Genre=F	Genre=F

Question n°2 :

Cette procédure permet de **fusionner** les deux tableaux H et F en **un tableau P** d'une manière **alternée** jusqu'à la fin du tableau ayant la plus petite taille ; **le reste des valeurs de l'autre tableau seront placées successivement à la fin de P.**

Question n°3 :

0) *Def Proc Traitement* (*n1, n2 : Octet ; H, F : Tab1 ; Var k : Octet ; Var P : Tab2*)

1) $k \leftarrow 0, i \leftarrow 0, j \leftarrow 0$

Répéter

$k \leftarrow k+1$

$i \leftarrow i+1$

$j \leftarrow j+1$

Si $H[i].id < F[j].id$ *Alors*

$P[k] \leftarrow H[i]$

$j \leftarrow j-1$

Sinon

$P[k] \leftarrow F[j]$

$i \leftarrow i-1$

Fin Si

Jusqu'à ($i=n1$) *ou* ($j=n2$)

Si ($i=n1$) *Alors*

Pour c *de* $j+1$ *à* $n2$ *Faire*

$k \leftarrow k+1$

$P[k] \leftarrow F[c]$

FinPour

Sinon

Pour c *de* $i+1$ *à* $n1$ *Faire*

$k \leftarrow k+1$

$P[k] \leftarrow H[c]$

FinPour

FinSi

2) *Fin Traitement*

Exercice 2 : (3 points)

Question n°1 :

$F(2, 2) = F(4, 1) = 4 * F(16, 0) = 4 * 1 = 4$

$F(2, 3) = 2 * F(4, 1) = 2 * 4 * F(16, 0) = 2 * 4 * 1 = 8$

Question n°2 :

Cette fonction retourne la **puissance d'ordre n** d'un entier a (a^n).

Question n° 3 :

- 0) Def FN F (a, n : Entier) : Entier Long
 1) Si n = 0 Alors F ← 1
 Sinon Si n mod 2 = 0 Alors F ← FN F (a*a, n div 2)
 Sinon F ← a * FN F (a*a, (n-1) div 2)
 FinSi
 2) Fin F

Exercice 3 : (4 points)**L'algorithme de la fonction Brun :**

- 0) Def FN Brun (Epsilon : Réel) : Réel
 1) B ← 0, k ← 1
 Répéter
 k ← k+2
 Si (FN Premier(k)) et (FN Premier (k+2)) Alors
 B1 ← B
 B ← B+1/k + 1/(k+2)
 FinSi
 Jusqu'à abs (B-B1) < Epsilon
 2) Brun ← B
 3) Fin Brun

Tableau de déclaration des objets

Objet	Type / Nature	Rôle
K	Entier Long	Compteur d'entier impair
B, B1	Réel	Calculer la constante de Brun
Premier	Fonction	Vérifier si un entier est premier

L'algorithme de la fonction Premier :

- 0) Def Fn Premier (N : Entier Long) : Booléen
 1) i ← 2
 TANTQUE (N mod i <> 0) ET (i ≤ N div 2) FAIRE
 i ← i+1
 FinTantque
 2) Premier ← (i > N div 2) ET (N > 1)
 3) Fin Premier

Tableau de déclaration des objets

Objet	Type / Nature	Rôle
i	Entier Long	Compteur d'entier impair

Problème : (10 points)**Analyse du programme Principal :**

Résultat = G
 G = Associer (G, "Code.txt"),
 Proc Resultat(M,G)
 M = Proc FormationMatrice(F,M)
 F = Associer (F, "Source.txt")

Tableau de déclarations des nouveaux types

Type
Matrice = Tableau de 40 x 40 chaîne [6]

Tableau de déclarations des objets globaux

Objet	Type / Nature	Rôle
F	Texte	Fichier texte à crypter
G	Texte	Fichier crypté
M	Matrice	Matrice utilisée pour crypter F
Resultat	Procédure	Permet de générer le fichier crypté à partir de la matrice M
FormationMatrice	Procédure	Permet de remplir la matrice M à partir du fichier à crypter

Algorithme de la procédure FormationMatrice :

- 0) Def Proc FormationMatrice (Var F : Texte ; Var M : Matrice)
- 1) Pour L de 1 à 40 Faire
 - Pour C de 1 à 40 Faire
 - M[L,C] ← "FFFFFF"
 - FinPour
- FinPour
- 2) Ouvrir(F), L ← 0
 - Tant que Non (Fin-Fichier(F)) Faire
 - Lire_nl(F,Lig)
 - L ← L+1
 - Si Long(Lig) mod 3 = 1 Alors Lig ← Lig + " _"
 - Sinon Si Long(Lig) mod 3 = 2 Alors Lig ← Lig + " _"
 - FinSi
 - C ← 0
 - Répéter
 - C ← C+1
 - M[L,C] ← Fn Hexa(Ord(Lig[1])) + Fn Hexa(Ord(Lig[2])) + Fn Hexa(Ord(Lig[3]))
 - Efface (Lig, 1, 3)
 - Jusqu'à Long (Lig)=0
 - FinTantque
- 3) Fermer(F)
- 4) Fin FormationMatrice

Tableau de Déclarations des Objets Locaux

Objet	Type / Nature	Rôle
L	Octet	Compteur de lignes
C	Octet	Compteur de colonnes
Lig	Chaîne de caractères	Contient une ligne du fichier à crypter
Hexa	Fonction	Calculer l'équivalent hexadécimal d'un entier de deux chiffres

Algorithme de la fonction Hexa :

- 0) Def Fn Hexa(k : Octet) : Chaîne
- 1) a ← k div 16
- 2) b ← k mod 16
- 3) Si a < 10 Alors Cha ← Chr (ORD ("0") + a)
 - Sinon Cha ← Chr (ORD ("A") + a - 10)
- FinSi
- 4) Si b < 10 Alors Chb ← Chr (ORD ("0") + b)
 - Sinon Chb ← Chr (ORD ("A") + b - 10)
- FinSi
- 5) Hexa ← Cha + Chb
- 6) Fin Hexa

Tableau de déclarations des objets locaux

Objet	Type / Nature	Rôle
a	Octet	Quotient de la division euclidienne d'un entier k Par 16
b	Octet	Reste de la division euclidienne d'un entier k Par 16
Cha	Caractère	Equivalent hexadécimal de a
Chb	Caractère	Equivalent hexadécimal de b

Algorithme de la procédure Résultat :

- 1) Def Proc Résultat (M : Matrice ; Var G : Texte)
- 2) Recréer(G)
 Pour C de 1 à 40 Faire
 Lig ← ""
 Pour L de 1 à 40 Faire
 Lig ← Lig + M[L,C]
 FinPour
 Ecrire_nl(G,Lig)
 FinPour
- 3) Fermer(G)
- 3) Fin Résultat

Tableau de déclarations des objets locaux

Objet	Type / Nature	Rôle
L	Octet	Compteur de lignes
C	Octet	Compteur de colonnes
Lig	Chaîne de caractères	Contient la concaténation du contenu d'une colonne