

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
 - 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