

EXERCICE 1 (3,25 points)

1) Résultats de la fonction Inconnue

Cas 1 : a = 8, b = 7

Appel	Condition	Action
Inconnue(8, 7)	7 impair	8 + Inconnue(8, 6)
Inconnue(8, 6)	6 pair	Inconnue(16, 3)
Inconnue(16, 3)	3 impair	16 + Inconnue(16, 2)
Inconnue(16, 2)	2 pair	Inconnue(32, 1)
Inconnue(32, 1)	1 impair	32 + Inconnue(32, 0)
Inconnue(32, 0)	b = 0	0

Résultat = 8 + 16 + 32 + 0 = **56** (= 8 × 7)

Cas 2 : a = 10, b = 6

Appel	Condition	Action
Inconnue(10, 6)	6 pair	Inconnue(20, 3)
Inconnue(20, 3)	3 impair	20 + Inconnue(20, 2)
Inconnue(20, 2)	2 pair	Inconnue(40, 1)
Inconnue(40, 1)	1 impair	40 + Inconnue(40, 0)
Inconnue(40, 0)	b = 0	0

Résultat = 20 + 40 + 0 = **60** (= 10 × 6)

2) Rôle de la fonction

La fonction **Inconnue(a, b)** calcule le **produit a × b** (multiplication par la méthode russe / décomposition binaire).

3-a) Tableau des erreurs

Ligne erronée	Correction
Ouvrir (ch2 , Fb ,"rb")	Ouvrir (ch2 , Fb ,"wb")
Tant que Fin_fichier (Ft) Faire	Tant que Non Fin_fichier (Ft) Faire

Ligne erronée	Correction
V.A ← Valeur(Sous_chaine (ch , 0 , p))	V.A ← Valeur(Sous_chaine (ch , 1 , p)) (ou indice 0 selon convention)
V.R ← Inconnue (A , B)	V.R ← Inconnue (V.A , V.B)
Ecrire (V , Fb)	Ecrire (Fb , V)

Remarque : Effacer(ch, 0, p+1) doit aussi être cohérent avec la convention d'indices (en commençant à 1 : Effacer(ch, 1, p)).

3-b) Tableau de déclaration des objets locaux

Procédure Remplir

Objet	Type/Nature
Ft	Texte
Fb	Fichier de Enreg
ch	Chaîne
p	Entier
V	Enreg

Tableau des nouveaux types

Type	Description
Enreg = Enregistrement	A : Entier ; B : Entier ; R : Entier ; Fin

EXERCICE 2 (2,75 points)

text

Procédure Tri (Ch1, Ch2 : Chaîne)

DEBUT

Ouvrir (Ch1, Fa, "rb")

n ← 0

Tant que Non Fin_fichier(Fa) Faire

Lire (Fa, T[n])

$n \leftarrow n + 1$

Fin Tant que

Fermer (Fa)

// Tri à bulles décroissant selon NSat

Répéter

permut \leftarrow Faux

Pour i de 0 à n-2 Faire

Si $T[i].NSat < T[i+1].NSat$ Alors

aux $\leftarrow T[i]$

$T[i] \leftarrow T[i+1]$

$T[i+1] \leftarrow$ aux

permut \leftarrow Vrai

FinSi

Fin Pour

Jusqu'à (permut = Faux)

Ouvrir (Ch2, Fc, "w")

Pour i de 0 à n-1 Faire

Ecrire_nl (Fc, $T[i].Comp + " "$ + Convch($T[i].NSat$)

+ " " + Convch($T[i].Nlnsat$))

Fin Pour

Fermer (Fc)

FIN

Déclarations

Objet	Type
Fa	Fichier de Compagnie

Objet	Type
Fc	Texte
T	Tableau [0..9999] de Compagnie
n, i	Entier
permut	Booléen
aux	Compagnie
Type	Description
Compagnie = Enregistrement	Comp : Chaîne ; NSat : Entier ; Nlnsat : Entier ; Fin

EXERCICE 3 (5 points)

text

Module NbreFort (N : Entier)

DEBUT

Ouvrir ("D:\Travail\NF.dat", F, "wb")

Pour Nb de 10 à N Faire

b ← SommeFacteurs(Nb)

Si (b ≤ 16) Alors

// Calcul de la somme des chiffres de Nb en base b

s ← 0

x ← Nb

Tant que (x > 0) Faire

s ← s + (x Mod b)

x ← x Div b

Fin Tant que

Si (s = b) Alors

E.Nbre ← Nb

E.Base ← b

Ecrire (F, E)

FinSi

FinSi

Fin Pour

Fermer (F)

FIN

Déclarations

Objet	Type
F	Fichier de NF
Nb, b, s, x	Entier
E	NF
Type	Description
NF = Enregistrement	Nbre : Entier ; Base : Entier ; Fin

Remarque clé : la somme des chiffres en base b est, par construction de la décomposition, la **somme des restes successifs** ($x \text{ Mod } b$). Pas besoin de convertir les chiffres en lettres : la somme numérique s'obtient directement.

EXERCICE 4 (9 points)

1) Procédure TrianglePascal

text

Procédure TrianglePascal (Var M : Mat ; N : Entier)

DEBUT

Pour L de 0 à N-1 Faire

$M[L, 0] \leftarrow 1$

$M[L, L] \leftarrow 1$

Pour C de 1 à L-1 Faire

$M[L, C] \leftarrow M[L-1, C] + M[L-1, C-1]$

Fin Pour

Fin Pour

FIN

2) Procédure Remplir

Sur la diagonale ascendante donnant le terme de rang i , les cases sont $M[i-1-k, k]$ pour k allant de 0 jusqu'à $(i-1) \text{ Div } 2$.

text

Procédure Remplir (M : Mat ; N : Entier ; Var TFS : Tab)

DEBUT

Pour i de 1 à N Faire

$sf \leftarrow 0$ // somme \rightarrow Fibonacci

$ni \leftarrow 0$ // nb impairs \rightarrow Stern

 Pour k de 0 à $(i-1) \text{ Div } 2$ Faire

$val \leftarrow M[i-1-k, k]$

$sf \leftarrow sf + val$

 Si $(val \text{ Mod } 2 = 1)$ Alors

$ni \leftarrow ni + 1$

 FinSi

 Fin Pour

$TFS[i].Fib \leftarrow sf$

$TFS[i].Ste \leftarrow ni$

Fin Pour

FIN

3) Programme principal

text

Algorithme Jeu

DEBUT

Répéter

Ecrire("Donner N : ")

Lire(N)

Jusqu'à ($N \geq 5$) ET ($N \leq 100$)

TrianglePascal(M, N)

Remplir(M, N, TFS)

Ouvrir("D:\Fidele.txt", FF, "r")

Ouvrir("D:\Gagnants.txt", FG, "w")

Tant que Non Fin_fichier(FF) Faire

Lire_nl(FF, mat)

// somme des chiffres du matricule

s ← 0

Pour j de 1 à Long(mat) Faire

s ← s + Valeur(mat[j])

Fin Pour

// recherche dans TFS

estFib ← Faux

estStern ← Faux

Pour i de 1 à N Faire

Si (TFS[i].Fib = s) Alors estFib ← Vrai FinSi

Si (TFS[i].Ste = s) Alors estStern ← Vrai FinSi

Fin Pour

Si (estFib ET estStern) Alors

 Ecrire_nl(FG, mat + " Super gagnant")

Sinon

 Si (estFib OU estStern) Alors

 Ecrire_nl(FG, mat + " Gagnant")

 FinSi

FinSi

Fin Tant que

Fermer(FF)

Fermer(FG)

FIN

Déclarations / Types

Type	Description
Mat = Tableau [0..99, 0..99] d'Entier	
Enreg = Enregistrement Fib : Entier ; Ste : Entier ; Fin	
Tab = Tableau [1..100] de Enreg	
Objet	Type
M	Mat
TFS	Tab
N, i, j, s	Entier
FF, FG	Texte
mat	Chaîne
estFib, estStern	Booléen