

Exercice 1 (3,5 points)

Instruction	Validité de l'instruction	Justification (Si Faux)
Pour Cj de Mardi à Samedi Faire Va[Cj].Jrs ← Cj Fin Pour	V	
Ecrire (Fa, Avend.CodeArt)	F	On ne peut écrire dans le fichier Fa qu'une valeur d'une variable de
Lire (Avend.Jrs)	F	On ne peut pas lire une variable de type scalaire énuméré.
Test ← Fin_Fichier (Ft) = Faux	V	
Ecrire (Fe, R)	F	On ne peut pas écrire un réel dans un fichier d'entiers.
Test ← Va[Lundi] > Va[Jeudi]	F	On ne peut pas comparer directement deux enregistrements.

Exercice 2 (3,5 points)

a) Le tableau de déclaration des objets locaux de la procédure **Inconnue**.

Objet	Nature/Type	Rôle
L	Entier	Compteur
C	Entier	Compteur
P	Entier	Compteur

b) Le résultat retourné par cette procédure pour N = 4 est :

M		1	2	3	4
1	1				
2	1	1			
3	1	2	1		
4	1	3	3	1	

c) **Cette procédure permet de remplir une matrice M par les N premières lignes du triangle de Pascal.**

Exercice N°3 : (4 points)

0) Début Calcul_Cos

1) Répéter

Ecrire ("Donner un réel : "), Lire(x)

Jusqu'à $(x \geq -1)$ Et $(x \leq 1)$

2) Ecrire(Fn Calcul(x))

3) Fin Calcul_Cos

0) Def fn Calcul(x:Réel):Réel

1) $C \leftarrow 1$

$i \leftarrow 0$

$P \leftarrow 1$

$F \leftarrow 1$

Répéter

$C_preced \leftarrow C$

$i \leftarrow i+2$

$P \leftarrow -P * x * x$

$F \leftarrow F*i*(i-1)$

$C \leftarrow C+P/F$

Jusqu'à $Abs(C - C_preced) < 10^{-4}$

2) Calcul $\leftarrow C$

3) Fin Calcul

Problème (9 points)

Analyse du programme principal

Nom = Brailles

3) Résultat = Écrire(Fn Convertir(G,F))

1) $G = Associer(G, "Braille.Txt")$

2) $F = Associer(F, "Codes_Braille.Dat")$

Le tableau de déclaration des nouveaux types

Type
CodeBraille = Enregistrement
L : Caractère
Code : Chaîne[6]
Fin CodeBraille
Carte = Fichier de
CodeBraille

Le tableau de déclaration des objets globaux

<i>Objet</i>	<i>Type/Nature</i>	<i>Rôle</i>
<i>Convertir</i>	<i>Fonction</i>	<i>Qui retourne l'équivalent alphabétique du texte écrit en braille.</i>
<i>G</i>	<i>Texte</i>	<i>Un fichier texte contenant le texte à convertir</i>
<i>F</i>	<i>Carte</i>	<i>Un fichier typé contenant les lettres majuscules et leurs équivalents en brailles</i>

Analyse de la fonction Convertir

DEF FN Convertir(Var G :Texte ; Var F : Carte) :Chaine

Résultat = Convertir

2) Convertir ← Sous_chaine(Ch,1,Long(Ch)-1)

1) Ch=[Ouvrir(G), Ch ← ""]

Tant que Non(Fin_Fichier(G)) Faire

Lire_nl(G,Ligne)

Ch ← Ch + Fn DetMot(Ligne, F) + " "

Fin Tant que

Le tableau de déclaration des objets locaux

<i>Objet</i>	<i>Type/Nature</i>	<i>Rôle</i>
<i>Ch</i>	<i>Chaine</i>	<i>La chaine alphabétique équivalente à la conversion du texte en Braille</i>
<i>Ligne</i>	<i>Chaine</i>	<i>Une variable servant à sauvegarder les lignes du texte en braille</i>
<i>DetMot</i>	<i>Fonction</i>	<i>Servant à déterminer le mot équivalent à une ligne écrite en Braille</i>

Analyse de la fonction DetMot

DEF FN DetMot (Ligne :Chaine ; Var F : Carte) :Chaine

Résultat = DetMot

2) DetMot ← M

1) M =[d ← 1, M ← ""]

Tant que d < Long(Ligne) Faire

Ch ← Sous_chaine(Ligne,d,6)

M ← M + Fn RechercheLettre(Ch,F)

d ← d + 6

Fin Tant que

Le tableau de déclaration des objets locaux

<i>Objet</i>	<i>Type/Nature</i>	<i>Rôle</i>
<i>d</i>	<i>Entier</i>	<i>Compteur</i>
<i>Ch</i>	<i>Chaine</i>	<i>Une séquence de 6 caractères dans une ligne Braille</i>
<i>M</i>	<i>Chaine</i>	<i>Une variable servant à sauvegarder un mot qui représente l'équivalent d'une ligne du texte en braille</i>
<i>RechercheLettre</i>	<i>Fonction</i>	<i>Servant à retourner la lettre équivalente à une séquence de 6 caractères écrits en Braille</i>

Analyse de la fonction RechercheLettre

DEF FN RechercheLettre (Ch :Chaine ; Var F : Carte) : Caractère

Résultat = RechercheLettre

2) RechercheLettre ← Enreg.L

1) Enreg = [Ouvrir(F)]

Répéter

Lire(F,Enreg)

Jusqu'à Enreg.Code=Ch

Le tableau de déclaration des objets locaux

<i>Objet</i>	<i>Type/Nature</i>	<i>Rôle</i>
<i>Enreg</i>	<i>CodeBraille</i>	<i>Une variable enregistrement servant à stocker la lettre et son équivalent Braille.</i>