

Correction Session de contrôle

Partie I

Exercice1 :

- 1) Booléen
- 2) On accepte toute valeur v telle que $3.5 < v < 19.9$
- 3) Cette fonction permet de vérifier si un tableau est trié ou non

Exercice 2 :

Fonction VERIF (n : entier) : booléen

Résultat = Verif \leftarrow R

R= [vrai] Si du \notin ["00", "25", "50", "75"] Alors R \leftarrow faux

FinSi

du \leftarrow ch[long(ch)-1] + ch[long(ch)]

ch= Convch (n,ch)

TDO		
Objet	Type / Nature	Rôle
R	Booléen	Permet de déterminer le résultat retourné par la fonction
Ch,du	chaîne	Permet de convertir l'entier n en une chaîne

Exercice3 :

0) Début fonction CalculPi : réel

1) $i \leftarrow 0$; $\pi_2 \leftarrow 2$

répéter

$\pi_1 \leftarrow \pi_2$

$i \leftarrow i + 1$

$\pi_2 \leftarrow \pi_2 + (\text{puis}(i+1) * \text{carré}(\text{fact}(i)) / \text{fact}(2*i+1))$

jusqu'à $\text{abs}(\pi_2 - \pi_1) \leq 0.0001$

2) calculpi $\leftarrow \pi_2$

3) Fin CalculPi

0) Début Fonction puis (n:entier) : réel ;

1) $p := 1$;

2) Pour j de 1 à n faire

$p \leftarrow p * 2$

3) Puis $\leftarrow p$

Fin Puis

Partie II

1°) **Analyse du PP**

Résultat = Affichage

Affichage = Proc Affiche (diagonale)

diagonale = [Associer (diagonale, "c:\diagonal.txt")]Proc Remplir (n,m, diagonale)

(n,m)= Proc Remplissage(n,m)

Fin PP

Tableau de déclaration des nouveaux types

Type
TypeMat = tableau [1..10,1..10] d'entiers

Tableau de déclaration des objets globaux

Objet	Type / Nature	Rôle
diagonale	Texte	Fichier texte contenant par ligne les éléments d'une diagonale droite
N	Entier	Taille de la matrice
M	TypeMat	Matrice de taille (n x n)
Affiche	Procédure	Procédure qui permet d'afficher les lignes contenant 4 chiffres distincts au moins
Remplir	Procédure	Procédure qui permet de remplir le fichier texte diagonal
Remplissage	Procédure	Procédure qui permet de saisir un entier n et de remplir la matrice m

Algorithme du PP

- 0) Début PP
- 1) Proc Remplissage(n,m)
- 2) Associer (diagonale, "c:\diagonal.txt")
Proc Remplir (n,m, diagonale)
- 3) Proc Affiche (diagonale)
- 4) Fin PP

2°) Analyse de la procédure REMPLISSAGE

Procédure REMPLISSAGE (var n : entier ; var M : typemat)

Résultat = M

M = [] Pour c de 1 à n faire

M[1,c] ← random(9)+1

FinPour

Pour l de 2 à n faire

Pour c de 1 à n-l+1 faire

S ← 0

Pour cpt de c à n-l+2 faire

s ← s + M[l-1,cpt]

FinPour

M[l,c] ← s

FinPour

FinPour

N = [] Répéter

N = donnée ("introduire la taille de la matrice")

Jusqu'à n dans [5..10]

Fin Remplissage

TDO de la procédure REMPLISSAGE

Objet	Type / Nature	Rôle
C,l,cpt	Variable/entier	Compteurs
S	Variable/entier	Variable intermédiaire, calcul de la somme

Algorithme de la procédure REEMPLISSAGE

0) Procédure REEMPLISSAGE (var n : entier ; var M : typemat)

1) Répéter

Ecrire ("introduire la taille de la matrice")

Lire (n)

Jusqu'à n dans [5..10]

2) Pour c de 1 à n faire

$M[1,c] \leftarrow \text{random}(9)+1$

FinPour

Pour l de 2 à n faire

Pour c de 1 à n-l+1 faire

$S \leftarrow 0$

Pour cpt de c à n-l+2 faire

$s \leftarrow s + M[l-1, \text{cpt}]$

FinPour

$M[l,c] \leftarrow s$

FinPour

FinPour

3) Fin Remplissage

Analyse de la procédure REEMPLIR

Procédure REEMPLIR (n :entier ; M :typeMat ; var diag : texte)

Résultat = diag

diag=[Recréer(diag)] Pour i de 1 à n faire

$ch \leftarrow ''$

$k \leftarrow i$

Pour j de 1 à i faire

Convch($M[j,k]$,ch1)

$ch \leftarrow ch + ch1$

$k \leftarrow k-1$

FinPour

Ecrire_nl(diag,ch)

FinPour

Fermer(diag)

Fin remplir

TDO de la procédure REEMPLIR

Objet	Type / Nature	Rôle
i, j, k	Entiers	Compteurs
Ch	Chaîne	Permet de former le contenu d'une ligne du fichier diagonal.txt
nbch	Chaîne	Permet de transformer le contenu d'une case en une chaîne

Algorithme de la procédure REEMPLIR

0) Procédure REEMPLIR (n :entier ; M :typeMat ; var diag : texte)

1) Recréer(diag)

Pour i de 1 à n faire

$ch \leftarrow ''$

$k \leftarrow i$

Pour j de 1 à i faire

Convch($M[j,k]$,nbch)

$ch \leftarrow ch + nbch$

FinPour

Ecrire_nl(diag,ch)

FinPour

Fermer(diag)

2) Fin remplir

Analyse de la procédure AFFICHE

Procédure AFFICHE (diag :texte)

Résultat = affichage

```
Affichage = [Ouvrir(diag)]  Tant que non (fin_fichier(diag)) faire
                                Lire_nl (diag, ch)
                                Si FN DISTINCTS(ch) alors écrire (ch)
                                Finsi
                                Fin Tant que
                                Fermer(diag)
Fin affichage
```

TDO de la procédure AFFICHE

Objet	Type / Nature	Rôle
Ch	Chaîne	Permet de lire le contenu d'une ligne du fichier diagonal.txt
Distincts	Fonction	Permet de vérifier si une chaîne comporte au moins 4 chiffres distincts

Algorithme de la procédure AFFICHE

0) Procédure affiche (diag :texte)

1) Ouvrir(diag)

```
Tant que non (fin_fichier(diag)) faire
    Lire_nl (diag, ch)
    Si FN DISTINCTS(ch) alors écrire (ch)
Finsi
```

Fin Tant que

Fermer(diag)

2) Fin affichage

Analyse de la fonction DISTINCTS

Fonction DISTINCTS (ch :chaîne) : booléen

Résultat = distincts \leftarrow verif

Verif = [verif \leftarrow vrai] Si long(chdist) < 4 alors verif \leftarrow faux

FinSi

Chdist = [i \leftarrow 0 ; chdist \leftarrow ch[1]] répéter

i \leftarrow i+1

[j \leftarrow 0] répéter

j \leftarrow j+1

jusqu'à (ch[i] = chdist[j]) ou (j = long(chdist))

Si ch[i] \neq chdist[j] alors chdist \leftarrow chdist + ch[i]

FinSi

Jusqu'à (i=long(ch)) ou (long(chdist)=4)

FinDistincts

TDO de la fonction Distincts

Objet	Type / Nature	Rôle
i,j	Entiers	Compteurs
Chdist	Chaîne	C'est une chaîne qui contiendra les chiffres distincts de ch
verif	logique	Variable intermédiaire

Algorithme de la fonction distincts

0) Fonction DISTINCTS (ch :chaîne) : booléen

1) $i \leftarrow 0$; $chdist \leftarrow ch[1]$

Repéter

$i \leftarrow i+1$

$j \leftarrow 0$

Répéter

$j \leftarrow j+1$

Jusqu'à ($ch[i] = chdist[j]$) ou ($j = \text{long}(chdist)$)

Si $ch[i] \neq chdist[j]$ alors $chdist \leftarrow chdist + ch[i]$

FinSi

Jusqu'à ($i = \text{long}(ch)$) ou ($\text{long}(chdist) = 4$)

2) $verif \leftarrow \text{vrai}$

Si $\text{long}(chdist) < 4$ alors $verif \leftarrow \text{faux}$

FinSi

3) $distincts \leftarrow verif$

3) FinDistincts