

Exercices du chapitre 6

Sommaire

Exercices

01-* -Saisie et affichage de notes 1	2
02-* -Saisie et affichage de notes 2	2
03-* -Saisie et affichage de notes 3	2
04-* -Affichage d'un tableau de notes avec une valeur sentinelle	2
05-**- Saisie et affichage de notes d'étudiants pour plusieurs matières	2
06-**- Palindrome.....	2
07-**-Initialisation partielle d'un tableau à 2 dimensions	2
08-***-Date du lendemain.....	3
09-***-Structure contenant un tableau	3
10-***-Modélisation d'un système scolaire : classe, étudiant, matière	3

Corrigés

01-* -Saisie et affichage de notes 1	2
02-* -Saisie et affichage de notes 2	2
03-* -Saisie et affichage de notes 3	3
04-* -Affichage d'un tableau de notes avec une valeur sentinelle	4
05-**- Saisie et affichage de notes d'étudiants pour plusieurs matières	5
06-**- Palindrome.....	6
07-**-Initialisation partielle d'un tableau à 2 dimensions	7
08-**-Date du lendemain.....	8
09-***-Structure contenant un tableau	9
10-***-Modélisation d'un système scolaire : classe, étudiant, matière	10

01-*-Saisie et affichage de notes 1

Ecrire un algorithme qui permet la saisie des notes de 10 étudiants, puis l'affichage de toutes les notes. On ne fera pas de saisie contrôlée des notes.

02-*-Saisie et affichage de notes 2

Ecrire un algorithme qui permet la saisie du nombre de notes à saisir, puis des notes. On sait simplement que le nombre de notes ne peut dépasser 50. On affiche ensuite les notes.

03-*-Saisie et affichage de notes 3

Sans réécrire tout l'algorithme précédent, écrire la portion d'algorithme qui permettra d'afficher à la fin, la note maximale, la note minimale, la moyenne.

04-*-Affichage d'un tableau de notes avec une valeur sentinelle

Ecrire un algorithme qui permet l'affichage du tableau de notes de taille 10. Le tableau aura été initialisé en dur dans l'algorithme (c'est à dire qu'il est déjà rempli). Les notes sont stockées dans le début du tableau et la valeur -1 indique la fin (logique) du tableau de notes. Le tableau peut contenir 10 notes.

12	8	16	5	-1
----	---	----	---	----	-----	-----	-----	-----	-----

Affichage :

[12, 8, 16, 5,]

05**- Saisie et affichage de notes d'étudiants pour plusieurs matières

Il y a un certain nombre de matières à envisager dont le nombre ne peut pas dépasser 7. Ecrire un algorithme qui permet la saisie du nombre de matières, du nombre d'étudiants et des notes par matière et par étudiant, puis l'affichage de l'ensemble des notes ainsi que les moyennes par matière et étudiant.

06**- Palindrome

Un mot est un palindrome s'il s'écrit de la même façon si on l'écrit à l'envers. "kayak" et "laval" sont des palindromes. Ecrire un algorithme qui permet de déterminer si un mot, considéré comme un tableau de caractères, est un palindrome ou non. On considérera que le tableau, d'au plus 50 éléments, est initialisé dans l'algorithme.

07**-Initialisation partielle d'un tableau à 2 dimensions

Ecrire l'algorithme qui permet d'initialiser certaines parties d'un tableau d'entiers à 2 dimensions de taille 10 x 10. Pour simplifier, on supposera que les cases du tableau sont par défaut initialisées à 0. L'algorithme doit d'abord initialiser la ligne d'indice 6 à 1, la colonne d'indice 9 à 1 puis chacune des diagonales à 1. L'algorithme doit ensuite afficher le tableau. Les indices du tableau commencent à 1.

1	0	0	0	0	0	0	0	1	1
0	1	0	0	0	0	0	0	1	0
0	0	1	0	0	0	0	1	1	0
0	0	0	1	0	0	1	0	1	0
0	0	0	0	1	1	0	0	1	0
1	1	1	1	1	1	1	1	1	1
0	0	0	1	0	0	1	0	1	0
0	0	1	0	0	0	0	1	1	0
0	1	0	0	0	0	0	0	1	0
1	0	0	0	0	0	0	0	1	1

08-***-Date du lendemain

Ecrire un algorithme qui demande à l'utilisateur, avec les variables appropriées, la date d'un certain jour, puis qui calcule et affiche la date du lendemain.

Remarques :

- les dates sont supposées valides,
- une année est bissextile si et seulement si elle est divisible par 4 et non par 100 (1900 n'était pas bissextile) ou divisible par 400 (2000 sera bissextile).

Vous pourrez utiliser un tableau pour stocker le nombre de jours de chaque mois.

09-***-Structure contenant un tableau

On peut donc définir par exemple le type TabEntiers de la façon suivante :

Types

```
TabEntiers : enrg
  tab      : tableau[1..50] d'entiers
  nbElements : entier
finEnrg
```

Ecrire un algorithme qui utilise la structure précédente et qui permet d'ajouter (à la suite des éléments déjà présents), des éléments au tableau. On posera la question "encore (O/N) ?" après chaque saisie.

10-***-Modélisation d'un système scolaire : classe, étudiant, matière

Créez la structure nommée "Matiere" qui permet de stocker les informations concernant une matière :

nom matière, coefficient de la matière.

En vous inspirant du 01, créez la structure nommée "Matières" qui permet de stocker les informations concernant l'ensemble des matières d'une classe (15 maximum).

Créez la structure nommée "Etudiant" qui permet de stocker les informations concernant un étudiant :

nom, prénom, ses notes correspondant à une note par matière (15 maximum), sa moyenne générale. On considèrera qu'un étudiant a une note dans chaque matière. On pourra définir un type "Notes" pour l'ensemble des notes.

En vous inspirant du 01, créez la structure nommée "Etudiants" qui permet de stocker les informations concernant l'ensemble des étudiants d'une classe (40 maximum).

Créez la structure nommée "Classe" qui permet de stocker les informations concernant une classe :

nom de la classe, l'ensemble des matières de la classes, l'ensemble des étudiants de la classe.

Ecrire un algorithme qui permet la saisie des matières d'une classe, puis la saisie de l'ensemble des informations concernant les étudiants.

CORRIGES

01-*-Saisie et affichage de notes 1

```

Algorithme SaisieAffichageNotes1
/* Déclarations */
Constantes
    MAX_NB_NOTES      = 50 /* le nombre maximal de notes */
Variables
    nbNotes    : entier /* nombre de notes à saisir */
    notes      : tableau[1..MAX_NB_NOTES] de réel /* les notes */
    i          : entier /* indice */
/* Instructions */
Début
    écrire("Saisie des notes")
    écrire ("combien de notes ? ")
    lire(nbNotes)
    pour i de 1 à nbNotes
        écrire ("note numéro ", i)
        lire(notes[i])
    finpour
    écrire("Affichage des notes")
    pour i de 1 à nbNotes
        écrire ("note numéro ", i, , " : ", notes[i])
    finpour
Fin

```

02-*-Saisie et affichage de notes 2

```

Algorithme SaisieAffichageNotes2
/* Déclarations */
Constantes
    MAX_NB_NOTES      = 50 /* le nombre maximal de notes */
Variables
    nbNotes    : entier /* nombre de notes à saisir */
    notes      : tableau[1..MAX_NB_NOTES] de réel /* les notes */
    i          : entier /* indice */
    reponse    : chaîne /* réponse oui/non */
/* Instructions */
Début
    reponse ← "oui"
    i ← 0
    écrire("Saisie des notes")
    tantque reponse = "oui" et i < MAX_NB_NOTES
        i ← i + 1
        écrire ("note numéro ", i)
        lire(notes[i])
        écrire ("encore une note ? (oui/non) ")
        lire(reponse)
    fintantque
    nbNotes ← i
    écrire("Affichage des notes")
    pour i de nbNotes à 1 par pas de -1
        écrire ("note numéro ", i, " : ", notes[i])
    finpour
Fin

```

03-*-Saisie et affichage de notes 3

Algorithme SaisieAffichageNotes2

/ Déclarations */*

Constantes

```

MAX_NB_NOTES      = 50  /* le nombre maximal de notes */
MIN                = 0   /* borne inférieure des notes */
MAX                = 20   /* borne supérieure des notes */

```

Variables

```

nbNotes   : entier /* nombre de notes à saisir */
notes     : tableau[1..MAX_NB_NOTES] de réel /* les notes */
i         : entier /* indice */
reponse   : chaîne /* réponse oui/non */
max       : réel /* note maximale */
min       : réel /* note minimale */
moy      : réel /* moyenne des notes */

```

/ Instructions */*

Début

```

min ← MAX
max ← MIN
moy ← 0
reponse ← "oui"
i ← 0
écrire("Saisie des notes")
tantque reponse = "oui" et i < MAX_NB_NOTES
  i ← i + 1
  écrire ("note numéro ", i)
  lire(notes[i])
  /* Détermination du min et du max */
  si notes[i] > max alors
    max ← notes[i]
  sinon
    si notes[i] < min alors
      min ← notes[i]
    finsi
  finsi
  /* Cumul pour le calcul de la moyenne */
  moy ← moy + notes[i]
  /* Saisie de la réponse */
  écrire ("encore une note ? (oui/non) ")
  lire(reponse)
fintantque
nbNotes ← i
moy ← moy / nbNotes
écrire("Affichage des notes")
pour i de nbNotes à 1 par pas de -1
  écrire ("note numéro ", i, " : ", notes[i])
finpour
écrire("note minimale : ", min)
écrire("note maximale : ", max)
écrire("moyenne : ", moy)

```

Fin

04-* -Affichage d'un tableau de notes avec une valeur sentinelle

Algorithme AffichageTableauValeurSentinelle

/ Déclarations */*

Constantes

TAILLE = 10 */* la taille du tableau */*

SENTINELLE = -1 */* valeur sentinelle */*

Variables

notes : tableau[1..MAX_NB_NOTES] de réel =

{12, 8, 16, 5, -1, , , , , } */* les notes */*

i : entier */* indice */*

/ Instructions */*

Début

i ← 1

écrire("[")

tantque notes[i] <> SENTINELLE et i <= TAILLE

écrire (notes[i], ", ")

i ← i + 1

fintantque

écrire("]")

Fin

05-- Saisie et affichage de notes d'étudiants pour plusieurs matières**

Algorithme SaisieAffichageTableauNotes2Dim

```
/* Déclarations */
```

```
Constantes
```

```
  NB_MAX_ETUD   = 50      /* le nombre maxi d'étudiants */
  NB_MAX_MAT    = 7       /* le nombre maxi de matières */
```

```
Variables
```

```
  notes        : tableau[1..NB_MAX_ETUD][1..NB_MAX_MAT] de réels
                /* les notes */
  nbEtud       : entier   /* le nombre d'étudiants */
  nbMat        : entier   /* le nombre de matières,
                          donc de notes par étudiant */
  i            : entier   /* compteur de boucles étudiant */
  j            : entier   /* compteur de boucles matière */
```

```
/* Instructions */
```

```
Début
```

```
  /* saisies */
```

```
  ecrire("combien de matières (pas plus de 7)")
```

```
  répéter
```

```
    lire(nbMat)
```

```
  jusqu'à (nbMat > 0 et nbMat <= NB_MAX_MAT)
```

```
  ecrire("combien d'étudiants (pas plus de 50)")
```

```
  répéter
```

```
    lire(nbEtud)
```

```
  jusqu'à (nbEtud > 0 et nbEtud <= NB_MAX_ETUD)
```

```
  pour i variant de 1 à nbEtud
```

```
    écrire ("saisir les notes de l'étudiant n° ", i, " : ")
```

```
    pour j variant de 1 à nbMat
```

```
      écrire("notes n° ", j, " : ")
```

```
      lire(notes[i][j])
```

```
    fpour
```

```
  fpour
```

```
  // affichage
```

```
  ecrire("affichage des notes")
```

```
  pour i variant de 1 à nbEtud
```

```
    écrire ("Notes de l'étudiant n° ", i, " : ")
```

```
    pour j variant de 1 à nbMat
```

```
      écrire("notes n° ", j, " : ", notes[i][j])
```

```
    fpour
```

```
  fpour
```

```
Fin
```

06-- Palindrome**

Algorithme Palindrome

/* Déclarations */

Constantes

LONGUEUR_MAX = 50 /* la longueur maximum du mot */

Variables

mot : tableau[1..LONGUEUR_MAX] de caractere =

{ 'k', 'a', 'y', 'a', 'k', ... } /* le mot */

longueur : entier = 5 /* le nombre de caractères du mot */

i : entier /* compteur de boucles */

milieu : entier /* indice du milieu, partie entière de
longueur/2 -1 */

estUnPalindrome : boolean /* vrai si le mot est un palindrome */

/* Instructions */

Début

estUnPalindrome ← vrai

milieu ← longueur div 2

i ← 1

tantque estUnPalindrome et i <= milieu

si mot[i] <> mot[longueur + 1 - i] alors

estUnPalindrome ← faux

sinon

i ← i + 1

finsi

fintantque

// affichage

si estUnPalindrome alors

écrire("le mot est un palindrome")

sinon

écrire("le mot n'est pas un palindrome")

finsi

Fin

07--Initialisation partielle d'un tableau à 2 dimensions**

Algorithme InitialisationTableau2Dim

Constantes

TAILLE = 10

Variables

t : tableau[1..TAILLE][1..TAILLE] d'entiers /* le tableau */

i : entier /* compteur de boucles 1^{er} indice : ligne */j : entier /* compteur de boucles 2^{ème} indice : colonne */

Début

// les 1

pour i de 1 à TAILLE

t[i][i] ← 1 // première diagonale

t[TAILLE-1-i][i] ← 1 // deuxième diagonale

t[5][i] ← 1 // ligne 5

t[i][8] ← 1 // colonne 8

fpour

// affichage

pour i de 1 à TAILLE

// affichage de la ligne i

écrire("| ")

pour j de 1 à TAILLE

écrire(t[i][j] + " | ")

finpour

écrireLigne()

// fin d'affichage de la ligne i

finpourFin

08--Date du lendemain**

Algorithme DateLendemain

/ Lit le jour , le mois et l'année d'une date choisie par l'utilisateur et calcule la date du lendemain */*

/ Déclarations */*

Variables

```
nbJoursMois   : tableau[1..12] de entier =
                {31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31}
jour           : entier /* le numéro du jour dans le mois */
mois          : entier /* le numéro du mois dans l'année */
an            : entier /* l'année */
jourLend      : entier /* le numéro du jour du lendemain */
moisLend      : entier /* le numéro du mois du lendemain */
anLend        : entier /* l'année du lendemain */
```

Début

```
écrire ("jour : ")
lire(jour)
écrire ("mois : ")
lire(mois)
écrire ("année : ")
lire(an)
jourLend ← jour
moisLend ← mois
anLend ← an
/* année bissextile ? */
si (an div 4 = 0 et an div 100 <> 0) ou (an div 400 = 0) alors
    nbJoursMois[2] ← nbJoursMois[2] + 1
fini
si jour < NB_JOURS_MOIS[mois] alors
    jourLend ← jourLend + 1
sinon
    jourLend ← 1
    si mois <> 12 alors
        mois ← mois + 1
    sinon
        mois ← 1
        an ← an + 1
    fini
fini
écrire("le lendemain de ", jour, "/", mois, "/", an, " est "
      jourLend, "/", moisLend, "/", anLend)
```

Fin

09-*-Structure contenant un tableau**

```
Algorithme StructureTableau
  /* Déclarations */
Constantes
  TAILLE      = 50
Types
  TabEntiers = enrg
  t          : tableau[1..TAILLE] d'entiers
  nb        : entier
  finEnrg
Variables
  tEntiers   : TabEntiers
  i          : entier /* indice */
  reponse   : chaine /* réponse oui/non */

/* Instructions */
Début
  reponse ← "oui"
  tEntiers.nb ← 0
  écrire("Saisie des notes")
  tantque reponse = "oui" et i < TAILLE
    tEntiers.nb ← tEntiers.nbElements + 1
    écrire ("note numéro ", tEntiers.nb)
    lire(tEntiers.t[tEntiers.nb])
    écrire ("encore une note ? (oui/non) ")
    lire(reponse)
  fintantque
Fin
```

10-*-Modélisation d'un système scolaire : classe, étudiant, matière**

Algorithme SaisieClasse

/* Déclarations */

Constantes

NB_MAX_MATIERES = 15

NB_MAX_ETUDIANTS = 50

Types

/* une matiere */

Matiere = enrg

nom : chaine

coef : réel

finEnrg

/* l'ensemble des matieres */

Matieres = enrg

t : tableau[1..NB_MAX_MATIERES] de Matiere

nb : entier

finenrg

/* l'ensemble des notes d'un étudiant */

Notes = enrg

t : tableau[1..NB_MAX_MATIERES + 1] de réel

nb : entier

finenrg

/* un étudiant */

Etudiant = enrg

nom : chaine

prenom : chaine

lesNotes : notes

finEnrg

/* l'ensemble des étudiants */

Etudiants = enrg

t : tableau[1..NB_MAX_ETUDIANTS] de Etudiant

nb : entier

finenrg

/* la classe */

Classe = enrg

nom : chaine

lesMatieres : Matieres

lesEtudiants : Etudiants

finenrg

Variables

laClasse : Classe /* la classe */

i : entier /* indice */

j : entier /* indice */

somCoef : réel /*somme des coefficients */

```
/* Instructions */
```

```
Début
```

```
/* saisie de la classe */
```

```
/* saisie du nom */
```

```
écrire("nom de la classe")
```

```
lire(laClasse.nom)
```

```
/* saisie des matières */
```

```
somCoef ← 0
```

```
écrire("combien de matières ? ")
```

```
lire(laClasse.lesMatières.nb)
```

```
pour i de 1 à laClasse.lesMatières.nb
```

```
    écrire("nom de la matière numero ", i)
```

```
    lire(laClasse.lesMatières.t[i].nom)
```

```
    écrire("coefficient de la matière numero ", i)
```

```
    lire(laClasse.lesMatières.t[i].coef)
```

```
    somCoef ← somCoef + laClasse.lesMatières.t[i].coef
```

```
finpour
```

```
/* saisie des étudiants */
```

```
écrire("combien d'étudiants ? ")
```

```
lire(laClasse.lesEtudiants.nb)
```

```
pour j de 1 à laClasse.lesEtudiants.nb
```

```
    écrire("nom de l'étudiant numéro ", j)
```

```
    lire(laClasse.lesEtudiants.t[j].nom)
```

```
    écrire("prénom de l'étudiant numéro ", j)
```

```
    lire(laClasse.lesEtudiants.t[j].prenom)
```

```
    laClasse.lesEtudiants.t[j].lesNotes.nb ←
```

```
        laClasse.lesMatières.nb + 1
```

```
/* saisie des notes */
```

```
pour i de 1 à laClasse.lesMatières.nb
```

```
    écrire("note en ", laClasse.lesMatières.t[i].nom)
```

```
    lire(laClasse.lesEtudiants.t[j].lesNotes.t[i])
```

```
/* cumul pour la moyenne */
```

```
    laClasse.lesEtudiants.t[j].lesNotes.t[laClasse.lesMatières.nb
```

```
        + 1] ← laClasse.lesEtudiants.t[j].lesNotes.t[i] *
```

```
        laClasse.lesMatières.t[i].coef
```

```
finpour
```

```
/* calcul de la moyenne */
```

```
laClasse.lesEtudiants.t[j].lesNotes.t[laClasse.lesMatières.nb
```

```
    + 1] ←
```

```
    laClasse.lesEtudiants.t[j].lesNotes.t[laClasse.lesMatières.nb
```

```
    + 1] /somCoef
```

```
finpour
```

```
Fin
```