

CHAPITRE 3 : INSTRUCTIONS REPETITIVES TRAVAUX DIRIGES

EXERCICE 1 * Mécanisme de l'instruction répétitive TANTQUE

Simuler l'exécution de chacun des algorithmes suivants :

ALGORITHME Tanque1

VARIABLES

nb : ENTIER

DEBUT

nb ← 10

TANTQUE nb < 40 FAIRE

 ECRIRE (nb)

 nb ← nb + 10

FINTANTQUE

ECRIRE ('le nombre vaut ', nb)

FIN

ALGORITHME Tanque2

VARIABLES

nb : ENTIER

DEBUT

nb ← 10

TANQUE nb > 40 FAIRE

 ECRIRE (nb)

 nb ← nb + 10

FINTANTQUE

ECRIRE ('le nombre vaut ', nb)

FIN

EXERCICE 4 * Chargement d'un camion

Ecrire un algorithme qui pilote le chargement d'un camion. Le camion est caractérisé par une capacité, masse qu'il peut transporter sans être en surcharge, qui sera donnée par l'utilisateur avant le début du chargement. Puis des paquets arrivent pour être chargés dans le camion. Chaque paquet est caractérisé par sa masse entrée par l'utilisateur. Si un paquet peut être chargé sans mettre le camion en surcharge, l'algorithme donne l'ordre de chargement. Le chargement doit s'arrêter avant le premier paquet qui ferait dépasser la capacité du camion. Lorsque le chargement est terminé, l'algorithme doit afficher :

le nombre de paquets chargés dans le camion

la masse totale des paquets chargés dans le camion. On suppose que l'utilisateur donne des valeurs positives pour la capacité du camion et pour la masse de chaque paquet.

EXERCICE 5 * Mécanisme de l'instruction répétitive pour

Simuler l'exécution de l'algorithme ci-dessous, puis écrire un algorithme équivalent utilisant une répétitive TANTQUE

ALGORITHME Pour1

VARIABLE

nb : ENTIER /* nombre courant */

i : ENTIER /* variable de boucle */

DEBUT

nb ← 10

POUR i DE 1 A 4 FAIRE

Ecrire (nb)

nb ← nb + 5

FINPOUR

Ecrire ("nombre = ", nb)

FIN

EXERCICE 6 * Moyenne

Ecrire un algorithme qui demande à l'utilisateur une suite de valeurs positives, et qui, sans mémoriser ces valeurs, calcule après la fin de saisie leur moyenne. On écrira deux versions de cet algorithme :

Version1 : On demande à l'utilisateur, avant la saisie de la suite de nombres, combien de nombres va comporter la suite.

Version 2 : On ne pose pas de question préalable, mais l'utilisateur indique qu'il a terminé la saisie en entrant un nombre spécial ne pouvant pas faire partie de la suite, par exemple le nombre -1.

EXERCICE 7 * Affichage n°1

Affichage de n 'a', n étant saisi au préalable. Ecrire l'algorithme correspondant.

Exemple d'exécution

n : 16

aaaaaaaaaaaaaaaa

Jusqu'ici, dans les exercices abordés, la forme exacte de l'affichage n'a pas eu d'importance. On supposera que l'instruction "ECRIRE" affiche un texte à l'écran sans passer à la ligne. Pour Passer à la ligne, il suffira d'afficher un caractère spécial, une constante nommée CRLF (pour Carriage Return Line Feed). Exemple : écrire("ceci est", " du texte", CRLF) affiche ceci est du texte avec passage à la ligne écrire(CRLF) passe à la ligne suivante.

FIN D'EXO

EXERCICE 8 ** Minimum, maximum et leur rang

Ecrire un algorithme qui étant donné une suite de nombres entiers demandés à l'utilisateur, annonce, à l'issue de la saisie, le minimum (ou le maximum, ou les deux) de ces nombres et son (leur) rang(s) dans la suite. On supposera que les nombres saisis par l'utilisateur sont compris entre deux bornes fixées dans l'algorithme. Dans le premier cas : minimum et maximum sont fonction des bornes Dans le second cas : minimum et maximum sont fonction du premier nombre saisi.

EXERCICE 9 ** Saisie de notes

Ecrire un algorithme qui permet la saisie de plusieurs notes comprises entre 0 et 20, en contrôlant leur validité. Après chaque saisie valide, l'algorithme demande : Encore une note (o/n)? Si une note n'est pas valide, le message suivant sera affiché : Erreur, la note doit être comprise entre 0 et 20.

EXERCICE 10 ** Affichage n°2

Affichage de p lignes de n 'a', n et p étant saisis au préalable. Ecrire l'algorithme correspondant.

aaaaaaaaaaaaaaaa

aaaaaaaaaaaaaaaa

FIN D'EXO

EXERCICE 11 ** Affichage n°3

Affichage de 4 'a' par ligne, sauf éventuellement sur la dernière ligne. Le nombre total de 'a' est saisi au préalable. Ecrire l'algorithme correspondant.

| | | |
|-------------------------------|------------------------|---------------|
| Si n=10 aaaa aaaa aa | Si n=8 aaaa aaaa | Si n=3 aaa |
|-------------------------------|------------------------|---------------|

EXERCICE 12 ** Affichage n°4

Affichage n°4 : vous ne devez utiliser que la structure pour.

| | |
|---|--|
| Si on a saisi 4 aaaa aaa aa a | Si on a saisi 5 aaaaa aaaa aaa aa a |
|---|--|

EXERCICE 13 ** Affichage n°5

| | |
|--|----------------------------|
| Si on a saisi 10 1 2 3 4 5 6 7 8 9 10 | Si on a saisi 4 1 2 3 4 |
|--|----------------------------|

EXERCICE 14 ** Affichage n°6

| | |
|-------------------------------|--|
| Si on a saisi 5 2 4 6 8 10 | Si on a saisi 8 2 4 6 8 10 12 14 16 |
|-------------------------------|--|

EXERCICE 15 ** Affichage n°7

On ne gèrera pas l'alignement des nombres. Il suffira de les espacer de 4 caractères.

| | | |
|--|------------------------------------|--------------------------|
| Si on a saisi 10 1 2 3 4 5 6 7 8 9 10 | Si on a saisi 8 1 2 3 4 5 6 7 8 | Si on a saisi 3 1 2 3 |
|--|------------------------------------|--------------------------|

EXERCICE 16 * Affichage n°8**

| | |
|---|--------------------------------|
| Si on a saisi 4 1 2 3 4 5 6 7 8 9 10 | Si on a saisi 3 1 2 3 4 5 6 |
|---|--------------------------------|

EXERCICE 17 * Affichage n°9**

| | | | |
|--|---|--|--------------------------------|
| Si on a saisi 10 aa aa aa aa aa | Si on a saisi 13 aaa aaa aaa aa aa | Si on a saisi 5 a a a a a | Si on a saisi 3 a a a |
|--|---|--|--------------------------------|

EXERCICE 18 * Recherche de répétitions 1**

On entre au clavier une série de chiffres ; la fin des entrées est marquée par la saisie du chiffre 0. On veut afficher les répétitions :

Un chiffre répété n fois (saisi n + 1 fois consécutivement) sera affiché n fois. Vous pourrez utiliser l'opérateur de concaténation.

Pour la suite 3 4 4 8 5 5 5 6 9 1 1 1 1 7 3 3 5 5 0, on obtient : 4 5 5 1 1 1 3 5

EXERCICE 19 * Recherche de répétitions 2**

On entre au clavier une série de chiffres ; la fin des entrées est marquée par la saisie du chiffre 0. On veut afficher les répétitions :

Un chiffre n fois (saisi n + 1 fois consécutivement) sera affiché 1 fois. Vous pourrez utiliser l'opérateur de concaténation.

Pour la suite 3 4 4 8 5 5 5 6 9 1 1 1 1 7 3 3 5 5 0, on obtient : 4 5 1 3 5

FIN

CHAPITRE 6 : VARIABLES STRUCTUREES TRAVAUX DIRIGES

CORRECTION

EXERCICE 1 * Saisie et affichage de notes 1

Algorithme SaisieAffichageNotes1

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 */

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

EXERCICE 2 * Saisie et affichage de notes 2

Algorithme SaisieAffichageNotes2

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 */

Début

reponse ← "oui"

i ← 0

écrire("Saisie des notes")

tantque reponse = "oui" et i < MAX_NB_NOTES **faire**

 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

EXERCICE 3 * Saisie et affichage de notes 3

Algorithme SaisieAffichageNotes2

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 : chaine /* réponse oui/non */
max : réel /* note maximale */
min : réel /* note minimale */
moy : réel /* moyenne des notes */
```

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)
```

Fin tant que

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])

Fin pour

écrire("note minimale : ", min)

écrire("note maximale : ", max)

écrire("moyenne : ", moy)

Fin

EXERCICE 4 * Affichage d'un tableau de notes avec une valeur sentinelle

Algorithme AffichageTableauValeurSentinelle

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 */

Début

i ← 1
 écrire("[")
tantque notes[i] <> SENTINELLE et i <= TAILLE
 écrire (notes[i], ", ")
 i ← i + 1
Fin
 écrire("]")

Fin

EXERCICE 5 ** Saisie et affichage de notes d'étudiants pour plusieurs matières

Algorithme SaisieAffichageTableauNotes2Dim

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 */

Début

/* saisies */
 écrire("combien de matières (pas plus de 7)")
 répéter
 lire(nbMat)
 jusqu'à (nbMat > 0 et nbMat <= NB_MAX_MAT)
 écrire("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
 écrire("affichage des notes")
pour i variant de 1 à nbEtud **faire**

```

écrire ("Notes de l'étudiant n° ", i, " : ")
pour j variant de 1 à nbMat faire
    écrire ("notes n° ", j, " : ", notes[i][j])
finpour
finpour
  
```

Fin

EXERCICE 6 ** Palindrome

Algorithme Palindrome

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 */

Début

estUnPalindrome ← vrai

milieu ← longueur div 2

i ← 1

tantque estUnPalindrome et i <= milieu **faire**

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

EXERCICE 7 ** 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 1er indice : ligne */

j : **entier** /* compteur de boucles 2ème 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
  Finpour
Fin

```

EXERCICE 8 *** 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 */*

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
  Finsi
  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

Finsi

Finsi

écrire("le lendemain de ", jour, "/", mois, "/", an, " est "
jourLend, "/", moisLend, "/", anLend)

Fin

EXERCICE 9 *** Structure contenant un tableau

Algorithme StructureTableau

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 */

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

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

Algorithme SaisieClasse

Constantes

NB_MAX_MATIERES = 15

NB_MAX_ETUDIANTS = 50

Types

Matiere = enrg

nom : chaine

coef : réel

FinEnrg

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 : chaîne  
prenom : chaîne  
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 : chaîne  
lesMatières : Matières  
lesEtudiants : Etudiants
```

Finenrg

Variables

```
laClasse : Classe /* la classe */  
i : entier /* indice */  
j : entier /* indice */  
somCoef : réel /*somme des coefficients */
```

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.lesMatiere.nb

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

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

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

 laClasse.lesEtudiants.t[j].lesNotes.t[laClasse.lesMatiere.nb + 1] ←

 laClasse.lesEtudiants.t[j].lesNotes.t[i] * laClasse.lesMatiere.t[i].coef

Finpour

/* calcul de la moyenne */

laClasse.lesEtudiants.t[j].lesNotes.t[laClasse.lesMatiere.nb + 1] ←

laClasse.lesEtudiants.t[j].lesNotes.t[laClasse.lesMatiere.nb + 1] /somCoef

Finpour

Fin

FIN