

TD 4 : Les tableaux :

Ex 1.
Procédure ex 1 (A : Tableau [1..100] de réel,
B : tableau [1..100] de réel, n : entier, m : entier)

Variable
R : réel
i, j : entier

Debut
 écriture ("...")
 lire (R)
 i ← 1
 Tant que (A[i] ≤ R et i ≤ n) faire
 B[i] ← A[i]
 i ← i + 1
 Fin tant que
 B[i] ← R
 pour j de i + 1 à n + 1 faire
 B[j] ← A[j - 1]
 Fin pour
 m ← n + 1
Fin

j ← i + 1
Tant que (j ≤ n + 1) faire
 B[j] ← A[j - 1]
 j ← j + 1
Fin Tant que

Ex 2.

Procédure Ex 2 (T : tableau [1..100] d'entier,
N : entier, TP : tableau [1..100] d'entier,
var NM : entier, TN : tableau [1..10] d'entier,
var N2 : entier)

Variable

Debut

j ← 1, k ← 1
pour i de 1 à N faire
 si (T[i] > 0) alors TP[j] ← T[i]
 j ← j + 1
 sinon TN[k] ← T[i]
 k ← k + 1

Fin si
fin pour

fin pour

N1 ← j - 1

N2 ← k - 1

2 solution:

N1 ← 0; N2 ← 0

pour i de 1 à N faire

 si (T[i] > 0) alors

 N1 ← N1 + 1

 TP[N1] ← T[i]

 sinon

 N2 ← N2 + 1

 TN[N2] ← T[i]

Fin si

Fin pour

Exercice 3.

Fonction saisie-taille() : entier
variable

Dim : entier

Debut

 Répéter

 écriture ("donner la dimension")

 lire (dim)

 Jusqu'à (dim > 1 et dim ≤ 50)

 retourner (dim)

Fin

Procédure saisie-tab (T: tab, dim: entier)

variable

i: entier

debut

pour i de 1 à dim faire

ecrire ("donner T(", i, ")")

lire (T[i])

Fimpour

Fin

Procédure Tri (T: tab, N: entier)

variable

i: entier

debut

Repete

OK ← faux

pour i de 1 à N-1 faire

si (T[i] > T[i+1]) alors

Permuter (T[i], T[i+1])

~~Repete~~

OK ← vrai

fini
Fimpour

Procédure Fusion (T1: tab, T2: tab, TF: tab
N: entier, M: entier)

variables

i1, i2, IF: entier

Debut

i1 ← 1, i2 ← 1, IF ← 1

Tantque (i1 ≤ n et i2 ≤ M) faire

Tantque (T1[i1] < T2[i2]) faire

TF[IF] ← T1[i1]

IF ← IF + 1

i1 ← i1 + 1

Fin Tantque

Tantque (T2[i2] < T1[i1]) faire

TF[IF] ← T2[i2]

IF ← IF + 1

i2 ← i2 + 1

Fin Tantque

Tantque (i1 ≤ n) faire

TF[IF] ← T1[i1]

IF ← IF + 1

i1 ← i1 + 1

Fin Tantque

Tantque (i2 ≤ M) faire

TF[IF] ← T2[i2]

IF ← IF + 1

i2 ← i2 + 1

Fin Tantque

Fin

Algorithme ex. 3

TYPE

Tab: Tableau [1..50] d'entiers

variables

T1, T2, TF: tab

N, M, L: entier

Debut

~~N ← saisie-tab~~

N ← saisie-taille()

Saisie-tab(T1, N)

Afficher(T, N)

Tri(T, N)

Affiche(T, N)

M ← saisie-taille()

saisie-tab(T2, M)

Tri(T2, M)

L ← M + N

Fusion(T1, T2, TF, N, M)

Affiche(TF, L)

Fin

Exercice 4:

Procédure chargement (T: tab, N: entier)

variable

i: entier

Debut

pour i de 1 à N faire

Repete

ecrire ("donner le produit n: ", i)

lire (T[i])

Jusqu'à (T[i] > 0)

Fin pour

Fin

Fonction somme (T: tab, N: entier): réel

variable

S: réel

i: entier

Debut

S ← 0

pour i de 1 à N faire

S ← S + T[i]

Fin Pour

retourner (S)

Fin

Fonction Mini (x: réel, y: réel): réel

variable

Debut

si (x < y) alors retourner (x)

sinon

retourner (y)

Fin si

Fin

Fonction ideal (T1: tab, T2: tab, N: entier): réel

variable

S: réel

debut i: entier

S ← 0

pour i de 1 à N faire

S ← S + Mini (T1[i], T2[i])

Fin pour

retourner (S)

Fin

Procédure affiche (T1: tab, T2: tab, N: entier)

variable

Debut

ecrire ("Le prix de panier d'après

le 1^{er} supermarché: ", Somme (T1, N))

ecrire (" " " " " 2^{eme} " " ", Somme (T2, N))

ecrire ("Prix ideal: ", ideal (T1, T2, N))

Fin

Algorithme ex 4

TYPE

tab: tableau [1..20] de réel

variable

A, B: tab

N: entier

Debut

Repete

ecrire ("donner le nombre des produits du panier")

lire (N)

Jusqu'à (N > 1 et N < 20)

ecrire ("Supermarché 1")

chargement (A, N)

Exercice 8:

Procédure Somme-Matrice (M_1 : tableau
 $[1..100][1..100]$ d'entiers, n : entier, m :
entier, M_2 : Tableau $[1..100][1..100]$ d'entiers,
 M_3 : Tableau $[1..100][1..100]$ d'entiers)

variable

i, j : entier

Debut

pour i de 1 à n faire

pour j de 1 à m faire

$M_3[i][j] \leftarrow M_1[i][j] + M_2[i][j]$

Fin pour

Fin pour

Fin

Exercice 9:

procédure saisie-mat (var nb : entier)

variable

Debut

Repeten

ecrite ("donner un entier")

lire (nb)

Jusqu'à ($nb > 0$ et $nb \neq 40$)

Fin

Procédure chargement (M : Mat, n :
entier)

variable

i, j : entier

debut

pour i de 1 à n faire

pour j de 1 à m faire

ecrite ("Donner

$M["i"]["j"]$)

lire ($M[i][j]$)

Fin pour

Fin pour

Fin

Procédure manipulation (A : Mat,
 n : entier, m : entier, B : Mat,
 P : entier, C : Mat)

variable

k, i, j : entier

Debut

pour i de 1 à n faire

pour j de 1 à P faire

$C[i][j] \leftarrow 0$

pour k de 1 à m faire

$C[i][j] \leftarrow C[i][j] + A[i][k]$
 $+ B[k][j]$

Fin pour

Fin pour

Fin pour

Fin

Algorithme programme principale

Type

Mat: Tableau $[1..100][1..100]$

d'entier

variable

m, n : entier

A : Mat

Debut

ecrite ("Nombre de ligne A")

saisie-nb (m)

ecrite ("Nombre de colonne A")

saisie-nb (m)

chargement (A, n, m)

ecrite ("Nb des colonnes B")

saisie-nb (P)

chargement (B, m, P) / Multiplication
 (A, n, m, B, P, C)