

Atelier 03 : Opérations sur les tableaux

Exercice 01 : Nombre de moyennes ≥ 10

En utilisant les tableaux, écrire un algorithme qui permet la saisie d'une liste de n moyennes réelles et d'afficher le nombre des moyennes supérieures ou égales à 10. On suppose que $n \leq 100$

Solution :

Algorithme :

Algorithme Moy_sup_à_10

// M : indice supérieur du tableau

Constante M = 100

// Moy : tableau de M éléments réels

Variable Moy : tableau[1..M] des réels

i, n, k : entiers

Début

Ecrire("Donner la valeur de n : ")

Lire(n)

// Saisie des éléments du tableau

Pour i ← 1 jusqu'à n Faire

Ecrire ("Donner l'élément ", i, " du tableau")

Lire (Moy[i])

FinPour

// k : compte le nombre de moyennes supérieures ou égales à 10.

k ← 0

Pour i ← 1 jusqu'à n Faire

Si Moy[i] ≥ 10 alors

k ← k + 1

FinSI

FinPour

// Affichage du résultat

Ecrire("Le nombre de moyennes supérieures ou égales à 10 est ", k)

Fin

Remarque :

- On peut ajouter dans l'algorithme les instructions suivantes pour gérer les cas où l'utilisateur saisisse une valeur supérieure à 100 (indice supérieur du tableau) :

Si n > 100 alors

n ← M

FinSI

- Remarquer bien l'avantage de la constante M qui peut être modifiée facilement pour la maintenance des programmes.

Exercice 02 : Permutation des éléments d'un tableau

Ecrire un algorithme qui permute les éléments d'un tableau de 8 éléments en plaçant le dernier élément en premier et ainsi de suite, voir la figure suivante :

15	18	13	99	68	71	96	55
----	----	----	----	----	----	----	----

Tableau initial

55	96	71	68	99	13	18	15
----	----	----	----	----	----	----	----

Tableau après permutation

Solution :

Algorithme :

Algorithme Permutation_tableau

Variables T : Tableau[1..8] d'entiers

I, X, K : Entier

Début

Pour I ← 1 jusqu'à 8 Faire

 Ecrire ("Donner l'élément ", I, " du tableau")

 Lire (t[I])

FinPour

K ← 8

// Notez bien que I varie de 1 à 4 dans la boucle suivante :

Pour I ← 1 jusqu'à 4 Faire

 X ← T(K)

 T(K) ← T(I)

 T(I) ← X

 K ← K - 1

FinPour

Pour I ← 1 jusqu'à 8 Faire

 Ecrire(T[i])

FinPour

Fin

Exercice 03 : Maximum des moyennes d'une classe

Ecrire un algorithme qui permet la saisie des moyennes d'une classe de n étudiants et affiche le maximum.

Solution :

Algorithme :

```
Algorithme maximum_tableau
// M : indice supérieur du tableau
Constante M = 100
Variable Moy : tableau[1 .. M] de réels
    i, n : entier
    max : réel
Début
    Ecrire("donner le nombre d'étudiants")
    Lire(n)
    Si n > M alors
        Ecrire("Dépassement de capacité du tableau")
        n ← M
    FinSI
    Pour I ← 1 jusqu'à n faire
        Ecrire ("Entrer la moyenne ",i, " :")
        Lire(Moy[i])
    FinPour
    max ← Moy[1]
    Pour i ← 2 jusqu'à n faire
        Si Moy[i] > max alors
            Max ← Moy[i]
        FinSI
    FinPour
    Ecrire ("le maximum des moyennes est ", max)
Fin
```

Exercice 04 : Trier un tableau

Ecrire un algorithme qui permet la saisie de 4 moyennes et de les

classer selon l'ordre croissant.

Solution :

Algorithme :

```
Algorithme Trier_tableau
// N : indice supérieur du tableau
Constante N = 4
Variables Moy : tableau[1 .. N] de réels
           I, J : entier
           X : réel
Début
  // Saisie des éléments du tableau
  Pour I ← 1 jusqu'à N faire
    Ecrire ("Entrer la moyenne ",I, " :")
    Lire(Moy[I])
  FinPour
  // Tri des éléments du tableau
  Pour I ← 1 jusqu'à N-1 faire
    Pour J ← I + 1 jusqu'à N faire
      Si Moy[I] > Moy[J] alors
        X ← Moy[I]
        Moy[I] ← Moy[J]
        Moy[J] ← X
      FinSI
    FinPour
  FinPour
  // Affichage des éléments du tableau trié
  Pour I ← 1 jusqu'à N faire
    Ecrire (Moy[I])
  FinPour
Fin
```

Exercice 05 : Tableau à deux dimensions

Ecrire un algorithme qui permet :

- la saisie des notes d'une classe de 5 étudiants en 4 matières
- calcul et affiche la moyenne de chaque étudiant
- calcul et affiche la moyenne de la classe dans chaque matière
- calcul et affiche la moyenne générale de la classe.

Solution :

Algorithme :

Algorithme Tableau_2_dimensions

// N : nombre d'étudiants, M : nombre de matières

Constante N = 5, M = 4

// Notes[I,J] : note de l'étudiant I dans la matière J

// Moy_E[I] : Moyenne de l'étudiant I

// Moy_M[J] : Moyenne de la classe dans la matière J

// Moy_G : Moyenne générale de la classe

Variable Notes : tableau[1..N, 1..M] des réels

Moy_E : tableau[1..N] des réels

Moy_M : tableau[1..M] des réels

I, J : entiers

S, Moy_G : réels

Début

// Saisie des notes

Pour I ← 1 jusqu'à N Faire

 Pour J ← 1 jusqu'à M Faire

 Ecrire ("Donner la note de l'étudiant ", I, " dans la matière", J)

 Lire (Notes[I, J])

 FinPour

FinPour

// Calcul et affichage de la moyenne de chaque étudiant

Pour I ← 1 jusqu'à N Faire

 S ← 0

 Pour J ← 1 jusqu'à M Faire

 S ← S + Notes[I, J]

 FinPour

 Moy_E[I] ← S / M

FinPour

Pour I ← 1 jusqu'à N Faire

 Ecrire("La moyenne de l'étudiant ", I, " est ", Moy_E[I])

FinPour

// Calcul et affichage de la moyenne par matière

Pour J ← 1 jusqu'à M Faire

 S ← 0

 Pour I ← 1 jusqu'à N Faire

 S ← S + Notes[I, J]

 FinPour

 Moy_M[J] ← S / N

FinPour

Exercice 06 : Produit de deux matrices

Ecrire un algorithme qui calcule la matrice C qui est le produit de deux matrices A et B. A est de l'ordre $n*m$ et B est de l'ordre $m*k$.

Solution :

Exemple :

$$\begin{array}{ccc}
 & \text{A} & & \text{B} & & \text{C} \\
 \\
 1 & 1 & 1 & \mathbf{X} & 1 & 1 & = & 3 & 3 \\
 1 & 1 & 1 & & 1 & 1 & & 3 & 3 \\
 & & & & 1 & 1 & & &
 \end{array}$$

A est de l'ordre $2 * 3$, B est de l'ordre $3*2$ et C est de l'ordre $2*2$. la valeur d'un élément de la matrice C est définie comme suit :

$$C(i, j) = \sum_{r=1}^{r=m} A(i, r) * B(r, j)$$

Dans cet exemple $m = 3$.

Note :

Le produit de deux matrices exige que le nombre de colonnes de la première matrice soit égal au nombre de lignes de la deuxième matrice.

Algorithme :

Algorithme Produit_deux_matrices

Variable A: tableau [1 To 20, 1 To 20] de réels

B: tableau [1 To 20, 1 To 20] de réels

C: tableau [1 To 20, 1 To 20] de réels

i, j, r, n, m, k : entier

début

Ecrire("Donnez le nombre de lignes de la matrice A : ")

lire(n)

Ecrire("Donnez le nombre de colonnes de la matrice A : ")

lire(m)

Ecrire("Donnez le nombre de colonnes de la matrice B : ")

lire(k)

//Remplissage de la matrice A

Pour i ← 1 jusqu'à n faire

 Pour r ← 1 jusqu'à m faire

 Ecrire("Donnez l'élément A[" , i , " , " , r,"]"))

 Lire(A[i, r])

 FinPour

FinPour

//Remplissage de la matrice B

Pour r ← 1 jusqu'à m faire

 Pour j ← 1 jusqu'à k faire

 Ecrire("donnez l'élément B[" , r , " , " , j , "]"))

 Lire(B(r, j))

 FinPour

FinPour

//Calcul de la matrice produit C

Pour i ← 1 jusqu'à n faire

 Pour j ← 1 jusqu'à k faire

 C[i, j] = 0

 Pour r ← 1 jusqu'à m faire

 C[i, j] = C[i, j] + A[i, r] * B[r, j]

 FinPour

 FinPour

FinPour

//Affichage des éléments de la matrice produit C

Pour i ← 1 jusqu'à n faire

 Pour j ← 1 jusqu'à k faire

 Ecrire("C[" , i , " , " , j , "]") = " , C(i, j))

Exercice 07 : Tableau dynamique à deux dimensions

En se basant sur l'annexe 01 qui traduit les instructions algorithmique en Visual Basic (voir la fin de l'ouvrage), écrire un programme VB qui utilise un tableau dynamique à deux dimensions (matrice) puis redimensionner le tableau en demandant le nombre de lignes et de colonnes. Ajouter des instructions pour remplir et afficher le contenu de ce tableau.

Solution :

Programme VB :

```
Sub main()  
    Dim A() As Double  
    Dim i As Integer, j As Integer  
    Dim n As Integer, m As Integer  
    n = Val(InputBox("Donnez le nombre de lignes de la matrice A"))  
    m = Val(InputBox("Donnez le nombre de colonnes de la matrice A"))  
    ' Redimensionner la matrice A  
    ReDim A(n, m)  
    ' Remplissage de la matrice A  
    For i = 1 To n  
        For j = 1 To m  
            A(i, j) = Val(InputBox("Donnez l'élément A(" & i & ", " & j & "  
                ")"))  
        Next j  
    Next i  
    ' Affichage des éléments de la matrice A  
    For i = 1 To n  
        For j = 1 To m  
            MsgBox "A(" & i & ", " & j & ")= " & A(i, j)  
        Next j  
    Next i  
End Sub
```

Exercice 08 : Recherche d'une valeur dans un

tableau

Ecrire un algorithme qui cherche une valeur x dans un tableau t.

Solution :

Algorithme :

Algorithme recherche

Variante t : tableau[100] de réels

x : réel

trouvé : logique

n, i, j : entier

Début

Ecrire("Donner la taille du tableau")

Lire(n)

Si $n > 100$ alors

n ← 100

FinSI

// Remplissage du tableau

Pour i ← 1 jusqu'à n faire

Ecrire("Donner l'élément t[", i, "]")

Lire(t[i])

FinPour

// Saisie de la valeur recherchée

Ecrire("Donner la valeur recherchée")

Lire(x)

// Recherche dans le tableau

trouvé ← Faux

Pour i ← 1 jusqu'à n faire

SI t[i] = x alors

j ← i

trouvé ← Vrai

// sortir de la boucle Pour

Quitter Pour

FinSI

FinPour

Si trouvé = Vrai alors

Ecrire(" La valeur ", x, " est trouvée, elle est au rang ", j)

Sinon

Ecrire(" La valeur ", x, " n'existe pas ")

FinSI

Fin

Note : Autres exercices sur les tableaux

L'atelier 08 contient d'autres exercices sur les tableaux. Il aborde les algorithmes de tri et de recherche.