

Algorithmique et programmation procédurale - TD No 3

Tableaux - CORRIGE

Exercice 1.

Ecrire une procédure qui permet de retourner le plus grand et le plus petit élément d'un tableau des réels.

Corrigé

Procédure Extremes(A[n] : Tableau de Reel, S min : Reel, S max : Reel)

Variables i : Entier

Début

```
    min ← A[1]
    max ← A[1]
    Pour i ← 2 à n
        Si max < A[i] Alors
            max ← A[i]
        Finsi
        Si min > A[i] Alors
            min ← A[i]
        Finsi
    FinPour
```

Fin

Exercice 2

Ecrire une procédure qui permet de normaliser les valeurs d'un tableau, c'est-à-dire on remplace l'élément A[i] par $(A[i] - \min) / (\max - \min)$ où max et min sont respectivement la valeur du plus grand et du plus petit élément du tableau.

Corrigé

Procédure Normaliser(ES A[n]: Tableau de Réel)

Variables i : Entier, min, max : Réel

Début

```
    Extremes(A[n],min,max)
    Pour i ← 1 à n
        A[i] ← (A[i]-min)/(max-min)
    FinPour
```

Fin

Exercice 3. A partir de deux tableaux précédemment saisis, écrivez un algorithme qui calcule le schtroumpf des deux tableaux. Pour calculer le schtroumpf, il faut multiplier

chaque élément du tableau 1 par chaque élément du tableau 2, et additionner le tout. Par exemple si l'on a :

Tableau 1 :

4	8	7	12
---	---	---	----

Tableau 2 :

3	6
---	---

Le Schtroumpf sera :

$$3 * 4 + 3 * 8 + 3 * 7 + 3 * 12 + 6 * 4 + 6 * 8 + 6 * 7 + 6 * 12 = 279$$

Corrigé

Programme Schtroumpf

Variables i, j, N1, N2, S : **Entier**

T1[], T2[] : **Tableau de Entier**

Debut

...

On ne programme pas la saisie des tableaux T1 et T2.

On suppose que T1 possède N1 éléments, et que T2 en possède N2

...

S ← 0

Pour i ← 1 à N1

Pour j ← 1 à N2

 S ← S + T1[i] * T2[j]

FinPour

FinPour

Ecrire "Le schtroumpf est : ", S

Fin

Exercice 4. Tableau dynamique

Ecrivez un algorithme permettant, à l'utilisateur de saisir les notes d'une classe en commençant par saisir le nombre d'étudiants. Le programme, une fois la saisie terminée, renvoie le nombre de notes supérieures à la moyenne de la classe.

Corrigé

Programme Note

Variables Nb, i, Nbsup : **Entier**, Som, Moy : **Réel**

T[] : **Tableau de Réel**

Debut

Ecrire "Entrez le nombre de notes à saisir : "

Lire Nb

T ← **CréerTableau**(Nb, Réel)

```

Pour i ← 1 à Nb
    Ecrire "Entrez la note n° ", i
    Lire T[i]
FinPour
Som ← 0
Pour i ← 1 à Nb
    Som ← Som + T[i]
FinPour
Moy ← Som / Nb
NbSup ← 0
Pour i ← 1 à Nb
    Si T[i] > Moy Alors
        NbSup ← NbSup + 1
    FinSi
FinPour
Ecrire NbSup, " élèves dépassent la moyenne de la classe"
Fin

```

Exercice 5. Ecrire une procédure qui permet de calculer le produit de 2 matrices qui sont représentées par 2 tableaux à deux dimensions.

Corrigé

Procédure Produit(S M3[m,p] : **Tableau de Réel**, M1[m,n], M2[n,p] : **Tableau de Réel**)

variables i, j, k : **Entier**

Début

```

    Pour i ← 1 à m
        Pour j ← 1 à p
            M3[i,j] ← 0
            Pour k ← 1 à n
                M3[i,j] ← M3[i,j] + M1[i,k].M2[k,j]
            FinPour
        FinPour
    FinPour
Fin

```

Exercice 6

Écrivez un algorithme qui permet de savoir si un tableau ne contient que des entiers de même signe.

Fonction memeSigne(T[n] : **Tableau d'Entier**) : **Booléen**

Variables i : **Entier**

Début

```

    Si (n = 1) Alors
        retourner vrai
    sinon // on a n ≥ 2
        Pour i ← 2 à n

```

```

                Si (T[1] * T[i] < 0) Alors
                    retourner faux
                FinSi
            FinPour
        retourner vrai
    FinSi
Fin

```

Exercice 7

Écrivez un algorithme qui permet de remplacer toutes les valeurs identiques contiguës dans un tableau d'entiers par une seule valeur. Par exemple : {1, 1, 1, 8, 2, 7, 7, 8, 9, 2} devient {1, 8, 2, 7, 8, 9, 2}.

Exercice 8

On appelle suite unimodale une suite d'entiers strictement positifs qui est :

- soit strictement croissante puis strictement décroissante ;
- soit strictement décroissante puis strictement croissante.

Par exemple, 1 2 5 6 1, 1 2 3 et 8 5 6 7 sont 3 suites unimodales alors que 1 3 2 4 et 8 5 5 6 7 ne le sont pas. Écrire une fonction booléenne estUnimodale qui détermine si une suite d'entiers donnée par un tableau est unimodale.

Corrigé

```

Fonction estUnimodale(T[n] : Tableau d'Entier) : Booléen
variables croissante, change, précédente : Booléen, i : Entier
Début
    Si (n = 1) Alors
        retourner vrai
    Sinon
        // on a n ≥ 2
        Si (T[1] = T[2]) Alors
            retourner faux
        Sinon
            change ← faux
            // on n'a pas encore atteint le premier
            // extremum local
            croissante ← (T[1] < T[2]) //ordre des 2 premiers éléments
            Pour i ← 2 à n-1
                précédente ← croissante
                Si (T[i] = T[i+1]) Alors
                    retourner faux
            Sinon
                croissante ← (T[i] < T[i+1])
                Si (croissante != précédente) Alors
                    Si (change) Alors
                        retourner faux
                    Sinon
                        change ← vrai
        FinSi

```

Fin
FinSi
FinSi
FinSi
FinSi
FinSi
FinPour
retourner vrai