

## Correction du T.D. 2

**Les tableaux****1 Exercice 1**

Ecrire les algorithmes permettant :

1. Le calcul du nombre d'occurrences d'un élément donné dans un tableau.

```
Nb_occurrences (T: Tableau d'entier, N: entier) : entier
VAR i,nb_occ : entiers
Debut
  nb_occ <- 0
  Pour i <- 1 a N Faire
    Si T[i] = X
      Alors nb_occ <- nb_occ + 1
    Fsi
  Fpour
  retourner nb_occ
Fin
```

2. Le calcul de la moyenne et du minimum des éléments d'un tableau.

```
Moyenne (T: Tableau d'entier, N: entier) : réel
VAR somme, i: entiers
  moyenne : réel
Début
  somme <- 0
  Pour i <- 1 a N Faire
    somme <- somme + T[i]
  Fpour
  moyenne <- somme / N
retourner moyenne
```

```
Minimum (T: Tableau d'entier, N: entier): entier
VAR min, i: entiers
Début
  min <- T[1]
  Pour i <- 2 a N Faire
    Si T[i]<min
      Alors min=T[i]
    Fsi
  Fpour
retourner min
```

3. De tester si un tableau est trié.

```

Est_trie (T: Tableau d'entiers, N: entier): booléen
VAR i: entiers
    est_trie: Booléen
Debut
    i <- 1
    Tant que i < N ET T[i] <= T[i+1] Faire
        i <- i + 1
    Ftque
    est_trie <- (i = N)
    retourner est_trie
Fin

```

4. Le calcul du produit scalaire de deux vecteurs réels  $u$  et  $v$  de dimension  $n$  :  $u.v = \sum_{i=1}^{i=n} u_i * v_i$

```

Produit_scalaire (u: Tableau d'entiers, v: Tableau d'entiers, n:
entier): entier
VAR i, prod_scalaire: entiers
Debut
    prod_scalaire <- 0
    Pour i <- 1 a n Faire
        prod_scalaire <- prod_scalaire + u[i] * v[i]
    Fpour
    retourner prod_scalaire;
Fin

```

## 2 Exercice 2

Ecrire l'algorithme effectuant le décalage des éléments d'un tableau.  
Exemple :

- Tableau initial 

D	E	C	A	L	A	G	E
---	---	---	---	---	---	---	---
- Tableau modifié (décalage à gauche) 

E	C	A	L	A	G	E	D
---	---	---	---	---	---	---	---

```

Procédure Decalage_gauche (T: Tableau de caractères, N: entier)
VAR tmp: caractère
    i: entier

```

```

Debut
    tmp <- T[1]
    Pour i <- 1 a N-1 Faire
        T[i] <- T[i+1]
    Ftque
    T[N] <- tmp
Fin

```

## 3 Exercice 3

Ecrire l'algorithme qui calcule le produit de deux matrices carrées réelles  $A = (a_{ij})$  et  $B = (b_{ij})$  de dimension  $n$  :  $c_{ij} = \sum_{k=1}^{k=n} a_{ik} * b_{kj}$ .

```

Produit_matriciel (a: Matrice carrée , b: Matrice carrée, n:
entier): Matrice carrée
VAR c: Matrice carrée n*n
    i: entier
Debut
    Pour i <- 1 a n Faire
        Pour j de 1 a n Faire
            c[i][j] <- 0
            Pour k de 1 a n Faire
                c[i][j] <- c[i][j] + a[i][k] * b[k][j]
            Fpour
        Fpour
    Fpour
    retourner c
Fin

```

## 4 Exercice 4

Soit un tableau  $T$  avec  $T(i) \in \{0,1\}$ . Ecrire un algorithme qui retourne la position  $i$  dans le tableau telle que  $T[i]$  est le début de la plus longue suite consécutive de zéros.

```
pos_suite_0 (t: Tableau d'entiers, n: entier): entier
```

```

VAR pos, lmax, lg, i: entiers
    suite: Booléen

Debut
    pos = -1
    lmax = 0
    suite = Faux
    pour i
    Pour i <- 1 a n Faire
        Si t[i]= 0
            Alors
                Si NON suite
                    Alors
                        lg <- 0
                        suite = 1
                Fsi
                lg = lg+1
            Sinon //t[i] différent de zéro
                Si suite = Vrai
                    Alors
                        suite <- Faux
                Si lg > lmax
                    Alors
                        lmax = lg
                        pos = i - lg
                Fsi
    Fsi

```

```

        Fsi
    Fsi
Fpour
Si suite=Vrai ET lg > lmax
    Alors
        pos = i - lg + 1
    Fsi
return pos
Fin

```

## 5 Exercice 5

Ecrire un algorithme qui calcule le plus grand écart dans un tableau (l'écart est la valeur absolue de la différence de deux éléments).

```

plus_grand_ecart (t: Tableau d'entiers, n: entier): entier
VAR: min, max, i: entiers
Debut
    min = t[1]
    max = t[1]
    Pour i <- 2 a n Faire
        Si t[i] > max
            Alors
                max = t[i]
        Fsi
        Si t[i] < min
            Alors
                min = t[i]
        Fsi
    Fpour
return max - min
Fin

```