

Digital Campus 2019/2020 – Algorithmie de la Data

NB: Dans les exercices suivants, on pourra utiliser la fonction `taille(tableau: Tableau[])` qui renvoie le nombre d'éléments dans le tableau

Tableaux

Exercice 0 :

Écrire un algorithme qui parcourt un tableau et qui affiche chacun des éléments.

```
Début
  i: Entier
  Pour i allant de 0 à taille(tableau)
    Afficher(tableau[i])
  FinPour
Fin
```

Exercice 1 :

Écrire un algorithme qui permet de permuter deux éléments du tableau.

Une solution possible :

```
Début
  n: Entier
  tableau: Tableau[n] d'Entiers
  tampon, element1, element2: Entier
  Si element1 < taille(tableau) ET element2 < taille(tableau) Alors
    tampon ← tableau[i1]
    tableau[i1] ← tableau[i2]
    tableau[i2] ← tampon
  Sinon
    Afficher("Valeur invalide pour les indices", element1, "ou", element2)
  FinSi
Fin
```

Ici on a utilisé une variable temporaire `tampon` pour stocker temporairement la première valeur afin que celle-ci ne soit pas perdue lors de la permutation.

Exercice 2 :

Écrire un algorithme qui renvoie le plus petit élément d'un tableau d'entiers.

Une solution possible :

```

Début
  n: Entier
  tableau: Tableau[n] d'Entiers
  min, i: Entier
  min ← tableau[0]
  Pour i allant de 1 à taille(tableau) par pas de 1 Faire
    Si min > tableau[i] Alors
      min ← tableau[i]
    FinSi
  FinPour
  Afficher(min)
Fin

```

Exercice 3 :

Écrire un algorithme qui, à partir d'un tableau d'entiers et d'un entier, affiche Vrai si l'entier est présent dans le tableau.

Une solution possible :

```

Début
  n: Entier
  tableau: Tableau[n] d'Entiers
  valeurRecherchee, i: Entier
  estPresente: Booléen
  Pour i allant de 0 à taille(tableau) par pas de 1 Faire
    Si valeurRecherchee == tableau[i] Alors
      estPresente ← VRAI
      i ← taille(tableau) # permet de sortir de la boucle
    FinSi
  FinPour
  Afficher(estPresente)
Fin

```

Une autre solution possible :

```

Début
  n: Entier
  tableau: Tableau[n] d'Entiers
  valeurRecherchee, i: Entier
  estPresente: Booléen
  estPresente ← FAUX
  i ← 0
  TantQue estPresente == FAUX ET i < taille(tableau) Faire
    estPresente ← valeurRecherchee == tableau[i]
    i ← i + 1
  FinTantQue
  Afficher(estPresente)
Fin

```

Exercice 4 :

Les notes d'un élève sont représentées par un tableau de réels. Écrire un algorithme qui permet de calculer la moyenne d'un élève.

Une solution possible :

```

Début
  n: Entier
  tableau: Tableau[n] de Réels
  moyenne: Réel
  i, somme: Entier
  somme ← 0
  Pour i allant de 0 à taille(tableau) par pas de 1 Faire
    somme ← somme + tableau[i]
  FinPour
  moyenne ← somme / taille(tableau)
  Afficher(moyenne)
Fin

```

Exercice 5 :

Écrire un algorithme qui permet d'afficher les éléments d'un tableau d'entiers à 2 dimensions

Une solution possible :

Début

nbLignes, nbColonnes: Entiers

tableau: Tableau[nbLignes][nbColonnes] d'Entiers

i, j: Entiers

Pour i allant de 0 à nbLignes par pas de 1 Faire

 Pour j allant de 0 à nbColonnes par pas de 1 Faire

 Afficher(tableau[i][j])

 FinPour

 Afficher("\n") // Retour à la ligne

FinPour

Fin