

# 1ALGO Algorithmique et logique

Gaël Roustan (Argonautes)

2025

**Abstract**

TD 3



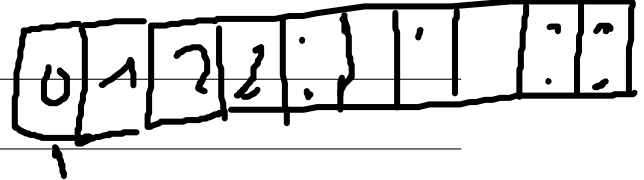
## TD 3

### Remplissage

Écrire un algorithme qui affecte à un tableau les dix premiers entiers de l'ensemble des entiers.

ALGORITHME : entiers de 0 à 9

CONSTANTES :



VARIABLES :

curseur : ENTIER

TABLEAU : mes\_dix\_entiers[0..9] : ENTIERS

DEBUT

POUR curseur ALLANT DE 0 A 9 AU PAS DE 1 FAIRE

mes\_dix\_entiers[curseur] <-- curseur

FINPOUR

FIN

### Recherche

Écrire un algorithme affichant si un réel saisi par l'utilisateur appartient ou non à un tableau de 10 réels positifs donné.

utilisateur alimente un tableau de 10 réels positifs

puis un autre utilisateur saisit un réel à tester

ALGORITHME: recherche

VARIABLES:

valeur\_recherchee : REEL

curseur : ENTIER

TABLEAU: reels\_pos[0..9] : REELS

DEBUT

POUR curseur ALLANT DE 0 A 9 AU PAS DE 1 FAIRE

REPETER

reels\_pos[curseur] <-- LIRE()

JUSQU'A reels\_pos[curseur] >= 0



```
    valeur_recherchee <-- LIRE()
    POUR curseur ALLANT DE 0 A 9 AU PAS DE 1 FAIRE
        SI valeur_recherchee = reels_pos[curseur] ALORS
            ECRIRE("valeur trouvée Youpi !!!")
        FINSI
```



```
    FINPOUR
```

```
    curseur <-- 0
```

```
    TANT QUE valeur_recherchee != reels_pos[curseur] ET cursor >= 0 ET cursor <= 8 A
```

```
        curseur <-- curseur + 1
```

```
    FINTANT QUE
```

```
    FIN
```

## Calcul des minimums et maximums

Écrire un algorithme qui affiche le plus petit et le plus grand élément d'un tableau de réels donné. Un seul « parcours » du tableau sera fait.

ALGORITHME: min\_max

demande à l'utilisateur de choisir la taille du tableau et le remplir

VARIABLES:

```
    TABLEAU: reels[0..4] <-- {12, 23.2, 7, 89, 94.32} : REELS
```

```
    curseur : ENTIER
```

```
    plus_petit, plus_grand : REELS
```

DEBUT

```
    plus_petit <-- reels[0]
```

```
    plus_grand <-- reels[0]
```

```
    POUR curseur ALLANT DE 1 A 4 AU PAS DE 1 FAIRE
```

```
        SI reels[curseur] < plus_petit ALORS
```

```
            plus_petit <-- reels[curseur]
```

```
        FINSI
```

```
        SI reels[curseur] > plus_grand ALORS
```

```
            plus_grand <-- reels[curseur]
```

```
        FINSI
```

```
    FINPOUR
```

```
    FIN
```

## Calcul des indices minimum et maximum

Écrire un algorithme qui trouve l'indice du plus petit élément et l'indice du plus élément d'un tableau de réels donné.

ALGORITHME: min\_max

VARIABLES:

TABLEAU: reels[0..4] <-- {12, 23.2, 7, 89, 94.32} : REELS

curseur : ENTIER

plus\_petit, plus\_grand : ENTIERS

DEBUT

plus\_petit <-- 0

plus\_grand <-- 0

POUR curseur ALLANT DE 1 A 4 AU PAS DE 1 FAIRE

SI reels[curseur] < reels[plus\_petit] ALORS

plus\_petit <-- curseur

FINSI

SI reels[curseur] > reels[plus\_grand] ALORS

plus\_grand <-- curseur

FINSI

FINPOUR

FIN

## Inversion d'un tableau

Écrire un algorithme qui réalise l'inversion d'un tableau d'entiers donné. Par exemple 1 2 3 4 5 devient 5 4 3 2 1. Nous n'utiliserons pas d'autre tableau que celui donné.

ALGORITHME : inversion

CONSTANTES :

indice\_max <-- 4 : ENTIER

VARIABLES :

TABLEAU: montableau[0..indice\_max] <-- {5, 4, 3, 2, 1} : ENTIERS

nouvelle\_position <-- indice\_max : ENTIER

valeur\_tmp : ENTIER

position : ENTIER

DEBUT

POUR position ALLANT DE 0 A 2 AU PAS DE 1 FAIRE

valeur\_tmp <-- montableau[indice\_max - position]

montableau[indice\_max - position] <-- montableau[position]

montableau[position] <-- valeur\_tmp

FINPOUR

ECRIRE(montableau)

FIN



## Somme de deux matrices

Écrire un algorithme qui calcule la somme  $S$  de deux matrices carrées  $A$  et  $B$  d'ordre  $n$ . Déclarer et initialiser la matrice  $A$ . Demander à l'utilisateur de saisir la matrice  $B$ .

ALGORITHME : somme matrices carrées

CONSTANTES :

$n \leftarrow 3$  : ENTIER

VARIABLES :

ligne, colonne : ENTIERS

TABLEAU :  $A[1..n][1..n] \leftarrow \{\{1, 2, 3\}, \{4, 5, 6\}, \{7, 8, 9\}\}$  : REELS

TABLEAU :  $B[1..n][1..n], S[1..n][1..n]$  : REELS

DEBUT

POUR ligne ALLANT DE 1 A  $n$  AU PAS DE 1 FAIRE

POUR colonne ALLANT DE 1 A  $n$  AU PAS DE 1 FAIRE

$B[ligne][colonne] \leftarrow \text{LIRE}()$

$S[ligne][colonne] \leftarrow A[ligne][colonne] + B[ligne][colonne]$

FINPOUR

FINPOUR

FIN

## Transposée d'une matrice

Écrire un algorithme qui calcule la transposée  $T$  d'une matrice à  $n$  lignes et  $p$  colonnes.

ALGORITHME : transposée

CONSTANTES :

$n \leftarrow 2$  : ENTIER

$p \leftarrow 3$  : ENTIER

VARIABLES :

TABLEAU :  $M[1..n][1..p], T[1..p][1..n]$  : REELS

ligne, colonne : ENTIERS

DEBUT

POUR ligne ALLANT DE 1 A  $n$  AU PAS DE 1 FAIRE

POUR colonne ALLANT DE 1 A  $p$  AU PAS DE 1 FAIRE

$M[ligne][colonne] \leftarrow \text{LIRE}()$

FINPOUR

FINPOUR

POUR ligne ALLANT DE 1 A  $p$  AU PAS DE 1 FAIRE

POUR colonne ALLANT DE 1 A  $n$  AU PAS DE 1 FAIRE

$T[ligne][colonne] \leftarrow M[colonne][ligne]$

FINPOUR

FINPOUR

FIN