

AP1 TD4 – Les tableaux : correction

Exercice 1 – L'algorithme mystère

```
ALGORITHME mystere
VARIABLES :
  Nb : tableau [1..5] d'entiers
  i : entier
DEBUT
  POUR i ALLANT DE 1 A 5 FAIRE
    Nb[i] ← i × i
  FINPOUR

  POUR i ALLANT DE 1 A 5 FAIRE
    afficher Nb[i]
  FINPOUR
FIN
```

1. Que produit l'algorithme ci-dessus ?

Il calcule les cinq premiers carrés qu'il place dans le tableau *Nb*, qui contient :

1	4	9	16	25
---	---	---	----	----

2. Peut-on simplifier cet algorithme avec le même résultat ?

```
ALGORITHME mystere
VARIABLES :
  Nb : tableau [1..5] d'entiers
  i : entier
DEBUT
  POUR i ALLANT DE 1 A 5 FAIRE
    Nb[i] ← i × i
    afficher Nb[i]
  FINPOUR
FIN
```

Exercice 2 – Un autre algorithme mystère

```
ALGORITHME mystere II
VARIABLES :
  Suite: tableau [1..8] d'entiers
  i : entier
DEBUT
  Suite[1] ← 1
  Suite[2] ← 1
  POUR i ALLANT DE 3 A 8 FAIRE
    Suite[i] ← Suite[i-1] + Suite[i-2]
  FINPOUR
FIN
```

Que contient le tableau *Suite* à l'issue de l'algorithme ?

1	1	2	3	5	8	13	21
---	---	---	---	---	---	----	----

Exercice 3 – Un algorithme fondamental : trouver le max d'un tableau

Soit un tableau d'entiers **T** à une dimension de taille 100. On suppose que **T** est déjà initialisé.

Ecrire un algorithme qui indique la valeur maximale du tableau **T**.

```
ALGORITHME max
VARIABLES :
  T: tableau [1..100] d'entiers
  i, max : entier
DEBUT
  max ← T[1]
  POUR i ALLANT DE 2 A 100 FAIRE
    SI T[i] > max ALORS
      max ← T[i]
    FINSI
  FINPOUR
  afficher max
FIN
```

Exercice 4 – La technique du flag

Soit un tableau **T** à une dimension de taille 100. On suppose que **T** est déjà initialisé.

Ecrire un algorithme qui demande à l'utilisateur une valeur et qui indique si cette valeur fait partie du tableau **T**.

Le flag, en anglais, est un petit drapeau, **représenté par une variable booléenne**, qui va rester baissé aussi longtemps que l'événement attendu ne se produit pas. Aussitôt que cet événement a lieu, le drapeau se lève (la variable booléenne change de valeur). Ainsi, la valeur finale de la variable booléenne permet au programmeur de savoir si l'événement a eu lieu ou non.

Dans cet exercice, le flag permet en outre de quitter la structure itérative dès qu'on a trouvé la valeur recherchée. On évite ainsi de parcourir l'intégralité du tableau.

La structure itérative utilisée est une boucle **REPETER JUSQU'A** car :

- on ne connaît pas le nombre d'itérations à réaliser : on en fera une dans le meilleur des cas (la valeur recherchée se situe dans la première case) et on en fera 100 si la valeur n'est pas dans le tableau

- on est certain de passer au moins une fois dans la boucle

On sort de cette boucle dans deux cas :

- soit on a trouvé la valeur
- soit, si elle n'existe pas, on a atteint la limite du tableau

ALGORITHME flag

VARIABLES :

T: tableau [1..100] d'entiers
i, val: entier
trouvé : booléen

DEBUT

saisir val

trouvé ← faux

i ← 1

REPETER

SI T[i] = val ALORS
| trouve ← vrai

FINSI

i ← i+1

JUSQU'A trouve = vrai OU i > 100

SI trouve = vrai ALORS

| afficher « valeur trouvée »

SINON

| afficher « valeur non trouvée »

FINSI

FIN

Exercice 5 – Tableau à deux dimensions

Soit un tableau *notes* à trente lignes et cinq colonnes représentant les moyennes des trente élèves d'une classe donnée dans cinq matières différentes. Le tableau *notes* est donc de la forme :

	Latin	Mathématiques	Dessin	Français	Histoire
Raoul	12	15,2	13	8,5	16
Julie	16	12,3	8	9,3	15
...

Ecrire un algorithme qui inscrit dans un tableau *moy* à une dimension les moyennes de classe pour chaque matière.

ALGORITHME moyenne

VARIABLES :

notes: tableau [1..30] [1..5] de réels
moy : tableau [1..5] de réels
matiere, eleve: entier
somme: réel

DEBUT

POUR matiere ALLANT DE 1 A 5 FAIRE

| somme ← 0

POUR eleve ALLANT DE 1 A 30 FAIRE

| | somme ← somme + notes[eleve] [matiere]

FINPOUR

| moy[matiere] ← somme / 30

FINPOUR

FIN

Attention à la notation : *notes[eleve] [matiere]* désigne la valeur du tableau *notes* située à la **ligne** numéro *eleve* et à la **colonne** numéro *matiere*.