

## Série de TDs N°4

### Exercice 1

**Algorithme** Algo\_1 ;  
**Var** i,N : entier ;  
Nb : tableau [1..50] de entier ;  
**Début**  
**Répéter**  
Lire (N);  
**Jusqu'à** ((N>0) et (N≤50)) ;  
  
**Pour** i ← 1 **jusqu'à** N **Faire**  
Nb[i] ← i \* i ;  
**FPour**  
  
**Pour** i ← 1 **jusqu'à** N **Faire**  
Ecrire (Nb[i]) ;  
**FPour**  
**Fin.**

**Algorithme** Algo\_2 ;  
**Var** i,N,S : entier ;  
Nb : tableau [1..50] de entier ;  
**Début**  
**Répéter**  
Lire (N);  
**Jusqu'à** ((N>0) et (N≤50)) ;  
  
S ← 0 ;  
**Pour** i ← 1 **jusqu'à** N **Faire**  
Nb[i] ← i \* i ;  
S ← S +Nb[i] ;  
**FPour**  
  
**Ecrire** (S) ;  
**Fin.**

**Algorithme** Algo\_3 ;  
**Var** i,N : entier ;  
T : tableau [1..80] de entier ;  
**Début**  
**Répéter**  
Lire (N);  
**Jusqu'à** ((N>0) et (N≤80)) ;  
T[1] ← 100 ;  
**Pour** i ← 2 **jusqu'à** N **Faire**  
T[i] ← T[i-1] + 2 ;  
**FPour**  
  
**Pour** i ← 1 **jusqu'à** N **Faire**  
Ecrire (T[i]) ;  
**FPour**  
**Fin.**

**Algorithme** Algo\_4 ;  
**Var** i,N : entier ;  
T : tableau [1..150] de entier ;  
**Début**  
**Répéter**  
Lire (N);  
**Jusqu'à** ((N>0) et (N≤150)) ;  
  
T[1] ← 8 ;  
**Pour** i ← 2 **jusqu'à** N **Faire**  
T[i] ← T[i-1] \* 3 ;  
**FPour**  
  
**Pour** i ← 1 **jusqu'à** N **Faire**  
Ecrire (T[i]) ;  
**FPour**  
  
**Fin.**

**Algorithme** Algo\_5 ;  
**Var** i,N,S,P : entier ;  
T : tableau [1..70] de entier ;  
**Début**  
**Répéter**  
Lire (N);  
**Jusqu'à** ((N>0) et (N≤70)) ;  
  
**Pour** i ← 1 **jusqu'à** N **Faire**  
Lire (T[i]) ;  
**FPour**  
S ← 0 ;  
P ← 1 ;  
**Pour** i ← 1 **jusqu'à** N **Faire**  
**Si** (T[i] mod 2 = 0) **Alors**  
S ← S + T[i] ;  
**Sinon**  
P ← P\*T[i] ;  
**FSi**  
**FPour**  
**Ecrire** (S,P) ;  
**Fin.**

**Algorithme** Algo\_6 ;  
**Var** i,N,j,k,X : entier ;  
T1 : tableau [1..40] de entier  
T2 : tableau [1..40] de entier  
**Début**  
**Répéter**  
Lire (N);  
**Jusqu'à** ((N>0) et (N≤40)) ;  
j ← 0 ; k ← 0 ;  
**Pour** i ← 1 **jusqu'à** N **Faire**  
**Lire** (X) ;  
**Si** (X mod 5 = 0) **Alors**  
j ← j+1 ;  
T1[j] ← X ;  
**Sinon**  
k ← k+1 ;  
T2[k] ← X ;  
**FSi**  
**FPour**  
  
**Pour** i ← 1 **jusqu'à** j **Faire**  
Ecrire (T1[i]) ;  
**FPour**  
  
**Pour** i ← 1 **jusqu'à** k **Faire**  
Ecrire (T2[i]) ;  
**FPour**  
**Fin.**

Soient les algorithmes précédents :

- 1- Faites la trace et dites que fait chacun des algorithmes ci-dessous.
- 2- Choisir l'un des algorithmes précédents et réécrire sa boucle **Répéter** par la boucle **TQue**.
- 3- Choisir l'un des algorithmes précédents et réécrire ses boucles **Pour** par des boucles **TQue**.

---

### Exercice 2

---

Pour calculer la moyenne d'un étudiant, écrire un algorithme en suivant les étapes suivantes :

- a. Lecture et sauvegarde de 8 notes de 8 modules dans un tableau dont leurs coefficients sont (2,3,1,2,4,2,1,3).
- b. Calcul et affichage de la moyenne de l'étudiant sans prise en compte les coefficients des modules.
- c. Calcul et affichage de la moyenne de l'étudiant en prenant en compte les coefficients des modules.

---

### Exercice 3

---

Soit TAB un tableau de N réels (N ne dépasse pas 120).

- 1- Ecrire un algorithme qui permet de vérifier si le tableau TAB est trié ou non.
- 2- Ecrire un algorithme qui permet de trier le tableau TAB dans l'ordre décroissant.
- 3- Ecrire un algorithme qui recherche la plus petite et la plus grande valeur ainsi que leurs positions dans le tableau TAB.
- 4- Ecrire un algorithme qui permet d'ajouter la valeur 15 à toutes les valeurs de rang pair et de soustraire la valeur 20 de toutes les valeurs de rang impair.
- 5- Ecrire un algorithme qui calcule la fréquence (le nombre de répétitions) du premier nombre du tableau, ainsi que toutes ses positions.

Exemple :

1.2	8.3	0	1.2	1.2	4.5	8	0	9	1.2	0.5	10	3	1.2	1.8
-----	-----	---	-----	-----	-----	---	---	---	-----	-----	----	---	-----	-----

Premier élément « 1.2 », sa fréquence = 5 (il existe 5 fois dans le tableau)

Ses positions sont : 1, 4, 5, 10, 14.

- 6- Ecrire un algorithme qui permet de recopier le tableau TAB dans un autre tableau TAB2 en supprimant les doublons.

Exemple

TAB	4.1	1.7	3	7	3	1.7	7	9.5	TAB2	4.1	1.7	3	7	9.5
-----	-----	-----	---	---	---	-----	---	-----	------	-----	-----	---	---	-----

- 7- Écrire un algorithme qui permet de scinder (diviser) TAB en deux tableaux TAB\_POS et TAB\_NEG contenant respectivement les nombres positifs et les nombres négatifs.

- 8- Ecrire un algorithme qui permet d'inverser les éléments d'un tableau.

Exemple

TAB	5	1	8	2	2	3	Son inverse est :	3	2	2	8	1	5
-----	---	---	---	---	---	---	-------------------	---	---	---	---	---	---

- 9- Ecrire un algorithme qui permet d'insérer dans le tableau TAB une valeur VAL à la  $k^{\text{ème}}$  position.  
**Remarque :** dans cette question ( $N < 120$ ).

- 10- On suppose que le tableau TAB est trié dans l'ordre croissant et ( $N < 120$ ). Écrire un algorithme qui insère un réel X dans ce tableau tout en conservant le tableau trié.

---

#### Exercice 4

---

Soient  $V1$  et  $V2$  deux vecteurs (tableaux à une dimension) de  $N$  entiers pour chacun ( $N \leq 150$ ).

- 1- Ecrire l'algorithme qui détermine la somme et produit scalaire de  $V1$  et  $V2$ .

La somme de  $V1$  et  $V2$  est  $V3$  tel que :  $V3[i] = V1[i] + V2[i]$   
Le produit scalaire de  $V1$  et  $V2 = \sum_{i=1}^N (V1[i] * V2[i])$ .

- 2- Ecrire un algorithme qui permet de construire, à partir des vecteurs précédents, deux vecteurs  $MAX$  et  $MIN$  définis comme suit :

- $MAX[i] = \text{maximum entre } V1[i] \text{ et } V2[i]$
- $MIN[i] = \text{minimum entre } V1[i] \text{ et } V2[i]$

- 3- On suppose que  $V1$  et  $V2$  sont triés dans l'ordre croissant :

- Ecrire un algorithme qui fusionne ces deux vecteurs en un vecteur  $V3$  trié dans l'ordre décroissant.
- Modifier l'algorithme précédent afin de supprimer les doublons éventuels.

---

#### Exercice 5

---

Soit  $T$  un tableau de caractères contenant une phrase se terminant par un point.

Écrire **un programme C** qui détermine :

- Le nombre de mots composant cette phrase
- Le plus long mot de cette phrase

---

#### Exercice 6

---

Soit  $Tab$  un tableau de  $N$  caractères et qui contient un seul mot.

- 1- Ecrire **un programme C** qui détermine le mot miroir d'un mot donné dans le tableau  $Tab$ .

Le mot "miroir", d'un mot donné, est obtenu en lisant ce dernier de la fin vers le début

Exemple :

Tab 

<b>B</b>	<b>a</b>	<b>T</b>	<b>n</b>	<b>A</b>
----------	----------	----------	----------	----------

      Son miroir est : 

<b>A</b>	<b>n</b>	<b>T</b>	<b>a</b>	<b>B</b>
----------	----------	----------	----------	----------

- 2- Ecrire **un programme C** qui vérifie si le mot donné dans le tableau  $Tab$  est un **palindrome**.

Un mot est dit palindrome, s'il est identique à son mot miroir.