

Module : Atelier de Programmation I **Responsable de cours:** Sassi Aymen
Responsables de TPs: Chakroun Rania, Hatem Ghorbel, Hssan Louati,
Manel Ammar, Nadia Chamkha, Nesrine Ksentini, Nesrine Triki, Rania
Boujelben

**Institut Supérieur
d'Informatique et
Multimédia de
Sfax**

A.U : 2022/2023

TP N° 4. Tableaux

Travail demandé

Exercice 1 :

Écrire les modules suivants :

1. **Saisie** : permet de remplir un tableau de réels (différents de 0) T de taille N ($1 \leq N \leq 50$) tel que : un élément quelconque ne peut apparaître dans T que trois fois au maximum.
2. **Décalage** : permet de supprimer les répétitions par un décalage à gauche et de remplacer la valeur supprimée par 0.
3. **Affichage** : permet d'afficher les éléments qui restent dans le tableau T (les 0 ne seront pas affichés).

Écrire un programme faisant appel à ces différents modules.

Exercice 2 Fusion de tableaux triés :

- 1) Écrire le module **FUSION ()** qui construit un tableau **FUS** trié par ordre croissant avec les éléments de deux tableaux A et B triés par ordre croissant. Pour deux tableaux de dimensions N et M, le tableau **FUS** aura la dimension N+M.

Méthode: utiliser trois indices i^a , i^b et i^{fus} . Comparer A [i^a] et B [i^b]; remplacer FUS [i^{fus}] par le plus petit des deux éléments ; avancer dans le tableau FUS et dans le tableau qui a contribué son élément.

Lorsque l'un des deux tableaux A ou B est épuisé, il suffit de recopier les éléments restants de l'autre tableau dans le tableau FUS.

Exercice 3 :

Écrire un programme en C qui permet de saisir N entiers ($0 < N < 50$) dans un tableau T. Le programme doit éclater ce tableau en deux tableaux TP et TN, TP doit contenir les entiers positifs ou nuls de T et TN les négatifs.

Exercice 4 :

1. Ecrire programme qui, étant donnés deux tableaux, détermine si le premier est inclus dans le second (si tous les éléments du premier tableau existent dans le second tableau).

A

2	8	4
---	---	---

B

4	5	8	9	2
---	-------	---	---	---	---

➔ Le tableau A est inclus dans B.

2. Soit un tableau d'entiers de dimension N, trié par ordre croissant (s'assurer qu'il est croissant au moment du remplissage). Inverser les éléments du tableau de manière à ce qu'il devienne décroissant (sans création d'un autre tableau).

B

4	5	18	19	22
---	---	----	----	----

Après inversion : B

22	19	18	5	4
----	----	----	---	---

Exercice 5 :

Développer les modules suivants :

- **Remplir** qui permet de remplir un tableau T par N entiers strictement positifs.
- **Afficher** qui permet d'afficher le tableau T qui contient N entiers.
- **Contact** qui permet de concaténer deux tableaux T1 et T2 de dimension respective N et M en un troisième tableau T3 sans copier les éléments qui se répètent plusieurs fois.
Contact doit calculer et retourner la dimension du tableau T3

Exemple :

T1:

5	3	2	7	2	7	3
---	---	---	---	---	---	---

T2:

10	7	5	10	8	5	4	9
----	---	---	----	---	---	---	---

T3:

5	3	2	7	10	8	4	9
---	---	---	---	----	---	---	---

Méthode :

Copier les éléments de T1 puis ceux T2 dans T3, avant d'insérer un «élément il faut vérifier s'il se trouve dans T3.

Vous utilisez la fonction **cherche ()** qui vérifie si l'entier X se trouve dans un tableau T de dimension D.

Écrire le programme C permettant d'appeler les fonctions définies ci-dessus et :

- De lire les dimensions respectives D1 et D2 de deux tableaux A et B ($2 < D1 < D2 < 20$).
- De remplir A et B par des entiers strictement positifs.
- De concaténer les tableaux A et B de dimensions respectives D1 et D2 en un troisième tableau C sans copier les éléments qui se répètent.
- D'afficher le tableau C.

Exercice 6 :

Écrire un programme qui remplit deux tableaux A et B avec des nombre positifs puis qui élimine la troisième occurrence du tableau B dans le tableau A.

Exercice 7 :

Créer des programmes en C qui permettent de chercher une valeur X (entrée au clavier) dans un tableau d'entiers avec les possibilités suivantes :

- 1) Déterminer si l'entier X appartient au tableau T
- 2) Compter le nombre d'occurrence de l'entier X.
- 3) La position de la première apparition de X
- 4) La dernière position d'apparition de X

Exercice 8 :

Écrire un programme qui permet de déterminer si une matrice est parfaite. Une matrice est parfaite si la somme de chaque ligne ou de chaque colonne donne la même valeur.

Exercice 9:

Soit un tableau T de taille N (avec $1 < N < 20$) qui contient un message binaire formé par des 0 ou 1, on veut vérifier la validité de ce message.

Un message est dit valide s'il contient au moins une séquence impaire de 1

Exemple :

0 1 1 1 0 0 1 1 0 0 → message valide

Écrire un programme qui remplit le tableau t puis affiche l'état du message (valide ou non valide)

Exercice 10 :

Écrire un programme qui permet de lire les dimensions de type entier ($1 \leq L \leq 50, 1 \leq C \leq 50$). Saisir au clavier les éléments du tableau M.

Créer et afficher un tableau à une dimension T_premiers qui contient tous les éléments premiers de la matrice M.

Exercice 11 :

Écrire un programme C qui calcule le maximum et le minimum des éléments d'un tableau Matrice d'entiers (m lignes et n colonnes).