

Algorithmique et programmation 1

Feuille d'exercices 4 - Tableaux

Avertissement : jusqu'alors en algorithmique on n'a pas trop géré les passages à la ligne dans les affichages. Comme dans ces exercices on a besoin de dire quand on passe à la ligne ou pas, je vous propose de supposer que :

- par défaut l'instruction **afficher** ne passe pas à la ligne
- si on veut passer à la ligne on utilise l'instruction **alaligne**

Exercice 1 _____ Affichage

Écrire un fragment d'algorithme (on ne demande pas d'écrire l'en-tête ni l'initialisation du tableau) qui affiche le contenu d'un tableau contenant n entiers en respectant le format suivant. Les valeurs sont séparées par une virgule, et l'affichage commence et termine par un crochet. Par exemple, un tableau contenant cinq entiers de valeurs 4, 5, 12, -3 et 0 donnera l'affichage [4, 5, 12, -3, 0].

Exercice 2 _____ Parcours d'un tableau

Dans cet exercice on considère que les tableaux T et T_{inv} contiennent des entiers et sont de dimension 100.

1. Écrire un algorithme qui réalise la saisie du nombre effectif n d'éléments puis la saisie des n entiers dans le tableau T (il faut s'assurer de ne pas saisir plus d'éléments que le tableau ne peut en contenir),
2. Écrire une fonction qui prend en paramètres un tableau et son nombre effectif d'éléments et calcule la somme des éléments d'indice pair du tableau T ,
3. Compléter votre algorithme pour qu'il affiche la somme calculée par votre fonction
4. Compléter encore l'algorithme pour qu'il remplisse un nouveau tableau T_{inv} avec les mêmes éléments que T mais dans l'ordre inverse. Exemple : si $T = [3, 5, -6, -2, 4, 3, 8]$ alors T_{inv} doit contenir $[8, 3, 4, -2, -6, 5, 3]$.

Exercice 3 _____ Décalages

Soit T un tableau d'entiers de dimension 100 initialisé et contenant n entiers (avec $0 \leq n \leq 100$). Écrire un algorithme qui demande répétitivement un choix à l'utilisateur (un entier entre 1 et 8) et selon ce choix effectue une des tâches suivantes (attention à bien mettre à jour la valeur de n quand c'est nécessaire) :

1. affiche le tableau (uniquement les n cases effectivement remplies)
2. demande à l'utilisateur un entier v et l'insère à la suite des éléments présents (ou message d'erreur si le tableau est plein),
3. supprime le dernier élément du tableau (ou message d'erreur si le tableau est vide),
4. demande à l'utilisateur des entiers v et k et remplace la valeur stockée à l'indice k par v (ou message d'erreur si k est hors du tableau ou hors des n valeurs effectives),
5. demande à l'utilisateur des entiers v et k et insère l'entier v à l'indice k (ou message d'erreur si problème),
6. demande à l'utilisateur un entier k et supprime la valeur d'indice k (ou message d'erreur si problème),

7. demande à l'utilisateur de choisir une direction (*dir*) entre gauche (g) et droite (d) et réalise une permutation circulaire simple (chaque élément est déplacé vers la case voisine).
Exemple :
Si $T = [3, 5, -6, -2, 4, 3, 8]$ et si $dir = "d"$ alors T doit devenir $[8, 3, 5, -6, -2, 4, 3]$.
Si $T = [3, 5, -6, -2, 4, 3, 8]$ et si $dir = "g"$ alors T doit devenir $[5, -6, -2, 4, 3, 8, 3]$
8. Sort de la boucle et s'arrête.

Exercice 4 _____ Parcours partiels

Dans la suite *Tab* désigne un tableau de dimension 100 contenant *nb* entiers. Écrire trois fonctions en pseudo-code, une pour chacune des tâches suivantes :

1. **est_la** prend trois paramètres : *nb*, *Tab* et un entier *v*, et retourne un booléen qui est **Vrai** si et seulement si la valeur *v* est présente dans *Tab*,
 2. **prem_occ** prend les mêmes paramètres et retourne l'indice de la première occurrence de *v* dans *Tab* si elle existe, -1 sinon,
 3. **sequence** prend les mêmes paramètres plus un entier *deb* et retourne l'indice auquel se termine la séquence croissante commençant à l'indice *deb* de *Tab* (*deb* est supposé $< nb$).
- Exemple : Si $Tab = [3, 5, -6, -2, 6, 9, 8]$ et $deb = 3$ alors le résultat est 5.

Exercice 5 _____ Réunion de deux tableaux

Supposons que l'on dispose de deux tableaux *T1* de dimension 50 contenant *n1* réels tous distincts, et *T2* de dimension 50 contenant *n2* réels tous distincts. Supposons qu'on a un troisième tableau, *T3* de dimension 100.

1. Écrire un algorithme qui copie dans *T3* toutes les valeurs de *T1* et toutes celles de *T2* (dans l'ordre que vous voulez).
2. Même question mais *T3* ne doit pas contenir de doublons. Il peut être utile de réutiliser une fonction de l'exercice précédent.

Exercice 6 _____ Tableaux à deux dimensions

Soit *Tab2dim* un tableau de dimension 20×20 contenant $n1 \times n2$ entiers (avec $n1 \leq 20$ et $n2 \leq 20$). Si vous le souhaitez, vous pouvez visualiser *Tab2dim* comme un tableau à *n1* lignes et *n2* colonnes. Écrire un algorithme qui va remplir les cases de *Tab2dim* par des entiers de la valeur 1 à la valeur $n1 \times n2$, en les parcourant dans l'ordre suivant (des exemples de résultats sont proposés à la fin) :

1. ligne par ligne, chaque ligne étant parcourue de droite à gauche, en commençant par la ligne du haut,
2. colonne par colonne, chaque colonne étant parcourue de bas en haut, en commençant par la dernière colonne,
3. diagonale par diagonale, en commençant par la diagonale ne contenant que le premier élément de la dernière ligne. Pour cette question vous pouvez, si vous le voulez, supposer que $n1=n2$

Cet exercice, sur un tableau de 3 par 5 cases donnerait le résultat suivant :

Par ligne :

```
5 4 3 2 1
10 9 8 7 6
15 14 13 12 11
```

Par colonne :

```
15 12 9 6 3
14 11 8 5 2
```

13 10 7 4 1

Par diagonale :

4 7 10 13 15

2 5 8 11 14

1 3 6 9 12

Exercice 7 _____ Palindrome

On rappelle que les différents caractères d'une chaîne de caractères peuvent être lus comme les cases d'un tableau. Écrire des fonctions en pseudo langage pour réaliser les tâches suivantes.

1. retourne **Vrai** si et seulement si son paramètre est un palindrome.
Exemple : "radar" est un palindrome.
2. retourne **Vrai** si et seulement si son paramètre est un palindrome quand on néglige les espaces.
Exemple : Si *texte* = "esope reste et se repose" alors *palindrome* vaut **Vrai**.
3. Écrire l'algorithme qui effectue la saisie d'une chaîne de caractères puis teste dessus vos deux fonctions.

Exercice 8 _____ Pour ceux qui ont fini

1. Reprendre la dernière question de l'exercice 2 mais en inversant les valeurs sans utiliser un tableau supplémentaire *Tinv*. Exemple : si $T = [3, 5, -6, -2, 4, 3, 8]$ alors *T* doit devenir $[8, 3, 4, -2, -6, 5, 3]$.
2. Re-répondre aux questions de l'exercice 6 mais maintenant une seule boucle est autorisée pour chaque question.
3. Re-répondre à la question 2 de l'exercice 4 mais cette fois *Tab* est trié par ordre croissant (on veut donc un algorithme beaucoup plus efficace),
4. Écrire un algorithme qui saisit deux entiers n et p tels que $0 \leq p \leq n$ et qui calcule le coefficient du triangle de Pascal : C_n^p . Pour cela, votre algorithme doit utiliser un unique tableau qui contient tour à tour toutes les lignes du triangle de Pascal jusqu'à la ligne n . Rappel : on a $C_n^0 = 1$ pour tout n , $C_n^k = 0$ pour tout n et tout $k > n$, $C_1^1 = 1$ et la formule pour calculer les autres coefficients est $C_n^p = C_{n-1}^{p-1} + C_{n-1}^p$.