

- Introduction à l'algorithmique et au langage C -

# Manipulation de tableaux

TP n°3

1<sup>re</sup> année ESIEA - Semestre 1

L. Beaudoin & R. Erra & A. Gademer & L. Avanthey

2016 - 2017

## Avant propos

Dans le TD « Introduction aux tableaux », nous avons appris les bases de l'utilisation des tableaux en langage C. Nous allons maintenant nous échauffer avec une série d'exercices un petit peu plus complexes.

## 1 Entrées/Sorties

### EXERCICE 1



Créez un fichier `mean.c` et écrivez-y le code du *Hello World*!. Vous supprimerez le `printf` après avoir testé la compilation.

### EXERCICE 2



Rajoutez la fonction `void loadArray(int size, int iArray[])` qui prend en paramètres un tableau et sa taille et qui remplit ce tableau avec des valeurs saisies au clavier.

- Les valeurs récupérées par le `scanf` sont au minimum aussi nombreuses que la taille du tableau (nous n'aurons pas de tableau à moitié rempli). Néanmoins, plus de valeurs que nécessaire peuvent être données. Vous serez donc attentif à ne pas affecter une valeur en dehors du tableau (indice en dehors de la taille), vous les ignorerez.
- Comme dans les TD et TP précédents, en cas de retour invalide du `scanf` vous afficherez `"Input error\n"` puis quitterez le programme avec l'appel de fonction `"exit(-1);"`.

### EXERCICE 3



Rajoutez la fonction `void showArray(int size, int iArray[])` qui prend en paramètres un tableau et sa taille et qui affiche un à un les éléments de ce tableau séparé par un unique espace. L'affichage se termine par un retour à la ligne.

### EXERCICE 4



Dans la fonction `main`, déclarez un tableau de dix entiers, puis appelez vos fonctions `loadArray` et `showArray`. Pensez à introduire l'appel à `loadArray` par un `printf`. Compilez et tester votre programme.

Voici un exemple de sortie attendue :

```
Please enter 10 integers:
9 6 10 7 9 3 5 12 -4 8
9 6 10 7 9 3 5 12 -4 8
```



### Attention !

Attention, le correcteur automatique compare la sortie au CARACTÈRE PRÈS. Respectez donc les espaces terminaux et les retours à la ligne (`\n`).



### Montrer l'invisible

Les caractères affichés en vert dans l'exemple de rendu correspondent à des caractères normalement **invisibles** : les espaces et les retour à la lignes. Ils suivent la table de correspondance suivante.

| Symbole | Signification                   |
|---------|---------------------------------|
| □       | représente un espace            |
| ↵       | représente un retour à la ligne |

## EXERCICE 5



Soumettez votre programme au correcteur automatique dans la section « Moyenne » (vous devez avoir deux exercices sur trois de validés).



### Attention !

Attention, le correcteur automatique se base sur le prototype pour reconnaître votre fonction ! Il est essentiel de respecter la casse (minuscule/majuscule) du nom et des paramètres de la fonction.

## 2 Calcul de la moyenne : parcours d'un tableau

### EXERCICE 6



Rajoutez la fonction `double getMean(int size, int iArray[])` qui prend en paramètres un tableau et sa taille et qui retourne la valeur de la moyenne des éléments de ce tableau.

### EXERCICE 7



Modifiez votre `main` pour afficher la valeur de la moyenne du tableau. Compilez et tester localement, puis soumettez au correcteur automatique.

Voici un exemple de sortie attendue :

```
Please enter 10 integers:
9 6 10 7 9 3 5 12 -4 8
9 6 10 7 9 3 5 12 -4 8
The mean of the elements equals: 6.500000
```

### 3 Extrémum, Indice de l'extrémum, Mode : recherche d'éléments dans un tableau

Une des recherches les plus courantes concerne la recherche de la valeur et de l'indice du maximum (du minimum) dans un tableau.

#### EXERCICE 8



Partant du programme précédent, copiez/collez les fonctions `loadArray` et `showArray` dans un fichier `minmaxmode.c`

#### EXERCICE 9



Rajoutez la fonction `int getMax(int size, int iArray[])` qui renvoie la valeur de l'élément maximum. Adaptez le `main` pour prendre en compte la nouvelle fonction, compilez, testez et soumettez.

#### EXERCICE 10



Rajoutez la fonction `int getMin(int size, int iArray[])` qui renvoie la valeur de l'élément minimum. Adaptez le `main` pour prendre en compte la nouvelle fonction, compilez, testez et soumettez.

#### EXERCICE 11



Rajoutez la fonction `int getMaxIndex(int size, int iArray[])` qui renvoie l'indice de l'élément maximum. En cas de multiples maxima, c'est l'indice du premier élément qui est renvoyé. Adaptez le `main` pour prendre en compte la nouvelle fonction, compilez, testez et soumettez.

#### EXERCICE 12



Rajoutez la fonction `int getMinIndex(int size, int iArray[])` qui renvoie l'indice de l'élément minimum. En cas de multiples minimums, c'est l'indice du premier élément qui est renvoyé. Adaptez le `main` pour prendre en compte la nouvelle fonction, compilez, testez et soumettez.

Voici un exemple de sortie attendue :

```
Please enter 10 integers:
9 6 10 12 9 3 5 12 -4 8
9 6 10 12 9 3 5 12 -4 8
The max of the elements equals: 12
The min of the elements equals: -4
The (first) max of the elements is at index: 3
The (first) min of the elements is at index: 8
```



#### Moyenne, Médiane, Mode

En statistiques, la moyenne, la médiane et le mode sont des critères de position, c'est-à-dire qu'ils fournissent des informations sur la répartition des valeurs dans un ensemble.

- La **moyenne** correspond à la valeur qu'aurait chacun des membres s'ils étaient tous identiques pour un total inchangé.
- La **médiane** correspond à la valeur qui permet de séparer l'ensemble en deux parties de même nombre d'éléments (autant d'éléments plus grands que d'éléments plus petits).
- Le **mode** correspond à la valeur la plus représentée dans l'ensemble (la valeur qui apparaît le plus grand nombre de fois).

Afin de calculer le mode, il nous faut calculer la fréquence d'apparition de chacun des éléments. Pour ce faire, nous allons créer un second tableau `occurrenceArray` dont les indices correspondront aux valeurs possibles dans le premier tableau et les valeurs au nombre de fois où celles-ci apparaissent effectivement.

Prenons par exemple le tableau suivant :

|                    |          |          |          |          |          |          |          |          |          |
|--------------------|----------|----------|----------|----------|----------|----------|----------|----------|----------|
| <b>int</b>         |          |          |          |          |          |          |          |          |          |
| <b>6</b>           | <b>5</b> | <b>2</b> | <b>6</b> | <b>2</b> | <b>3</b> | <b>5</b> | <b>2</b> | <b>7</b> | <b>8</b> |
| <i>myArray</i> [0] | [1]      | [2]      | [3]      | [4]      | [5]      | [6]      | [7]      | [8]      | [9]      |

Nous pouvons voir dans ce tableau que la valeur 2 apparaît 3 fois, que la valeur 5 et 6 apparaissent 2 fois et que les nombres 3, 7 et 8 apparaissent 1 fois.

Le tableau d'occurrence correspondant aux valeurs du tableau précédent donnera donc :

|                       |          |          |          |          |          |          |          |          |          |
|-----------------------|----------|----------|----------|----------|----------|----------|----------|----------|----------|
| <b>int</b>            |          |          |          |          |          |          |          |          |          |
| <b>0</b>              | <b>0</b> | <b>3</b> | <b>1</b> | <b>0</b> | <b>2</b> | <b>2</b> | <b>1</b> | <b>1</b> | <b>0</b> |
| <i>occurrence</i> [0] | [1]      | [2]      | [3]      | [4]      | [5]      | [6]      | [7]      | [8]      | [9]      |

Nous pouvons lire la case d'indice 2 de ce dernier en disant : "La valeur 2 apparaît 3 fois dans myArray".



#### Dans cet exercice

Nous considérerons arbitrairement que les éléments du tableau sont des nombres entiers entre 0 et 20.

### EXERCICE 13



Rajoutez la fonction `int getMode(int size, int iArray[])` qui renvoie le mode du tableau. Indice : après avoir calculé le tableau d'occurrence puis recherché l'indice du maximum de ce dernier. En cas de modes multiples, le plus petit est renvoyé. Adaptez le `main` pour prendre en compte la nouvelle fonction, compilez, testez et soumettez.

Voici un exemple de sortie attendue :

```
Please enter 10 integers:
6 5 2 6 2 3 5 2 7 8
6 5 2 6 2 3 5 2 7 8
The mode of the array equals: 2
```

## 4 Médiane : tri des éléments d'un tableau

### EXERCICE 14



Partant du programme `mean.c`, copiez/collez les fonctions `loadArray` et `showArray` dans un fichier `median.c`

### EXERCICE 15



Rajoutez la fonction `void swap(int iArray[], int index1, int index2)` qui échange les éléments d'indice `index1` et `index2` du tableau (algorithme d'échange que nous avons déjà croisé plusieurs fois maintenant). Adaptez le `main` pour prendre en compte la nouvelle fonction, compilez, testez et soumettez.

**EXERCICE 16**

Rajoutez la fonction `void bubbleSort(int size, int iArray[])` qui ordonne les éléments du tableau dans l'ordre croissant. L'algorithme du tri à bulle a été vu dans le TD6. Adaptez le `main` pour prendre en compte la nouvelle fonction, compilez, testez et soumettez.

**EXERCICE 17**

Rajoutez la fonction `int getMedian(int size, int iArray[])` qui renvoie la valeur médiane du tableau. Indice : dans un tableau ordonné, la valeur médiane est celle qui se trouve au milieu. En cas d'un nombre pair de cases, nous prendrons la case d'indice  $\text{taille}/2$ . Adaptez le `main` pour prendre en compte la nouvelle fonction, compilez, testez et soumettez.

Voici un exemple de sortie attendue :

```
Please enter 10 integers:
9 6 10 12 9 3 5 12 -4 8
-4 3 5 6 8 9 9 10 12 12
The median of the array equals: 9
```