



CPPLI : TD 2 : C : Tableau statique

Romain Absil Jonas Beleho Pierre Bettens
David Hauweele Pierre Hauweele
Nicolas Vansteenkiste * (ESI – HE2B)

Année académique 2021 – 2022

Ce TD¹ aborde les tableaux statiques du langage C dans sa version C11² (ISO/IEC 9899:2011).

Ex. 2.1 Écrivez la fonction de prototype :

```
void arrayIntPrint(const int data [], unsigned nbElem);
```

Elle affiche sur la sortie standard les `nbElem` premiers éléments du [tableau](#)³ d'`int` en argument, séparés par un espace. Un passage à la ligne termine l'affichage. Si `data` vaut `NULL`⁴ ou si le tableau contient moins de `nbElem` éléments, la fonction adopte un [comportement indéterminé](#)⁵.

Ex. 2.2 Écrivez la fonction de prototype :

*Et aussi, lors des années passées : Monica Bastreggi, Stéphan Monbaliu, Anne Rousseau et Moussa Wahid.

1. https://poesi.esi-bru.be/pluginfile.php/7311/mod_folder/content/0/td02_c/td02_c.pdf (consulté le 13 septembre 2021).

2. [https://en.wikipedia.org/wiki/C11_\(C_standard_revision\)](https://en.wikipedia.org/wiki/C11_(C_standard_revision)) (consulté le 13 septembre 2021).

3. <https://en.cppreference.com/w/c/language/array> (consulté le 13 septembre 2021).

4. <https://en.cppreference.com/w/c/types/NULL> (consulté le 13 septembre 2021).

5. https://en.wikipedia.org/wiki/Undefined_behavior (consulté le 13 septembre 2021).

```
void arrayIntSort(int data [], unsigned nbElem, bool ascending);
```

Elle trie les `nbElem` premiers éléments de `data` dans l'ordre croissant ou décroissant selon que le paramètre `ascending` soit `true` ou `false`⁶. Implémentez l'algorithme de tri⁷ de votre choix.

Testez votre fonction de tri et vérifiez son bon fonctionnement avec la fonction d'affichage de l'Ex. 2.1.

Ex. 2.3 À l'aide de la fonction standard `qsort`⁸, triez un tableau d'`int` dans l'ordre :

- (a) croissant ;
- (b) décroissant ;
- (c) croissant⁹ modulo 3.

Ex. 2.4 Reprenez votre fonction de tri de l'Ex. 2.2 et modifiez-la pour produire la fonction de prototype :

```
void arrayIntSortGeneric(int data [], unsigned nbElem,  
                          int (*comp)(const int *, const int *));
```

Celle-ci trie selon l'algorithme de votre choix les `nbElem` premiers éléments de `data` en utilisant la fonction `comp` pour comparer les éléments comme le fait la fonction standard `qsort`.

Testez votre fonction de tri et vérifiez son bon fonctionnement avec la fonction d'affichage de l'Ex. 2.1 et des fonctions de tri comme dans l'Ex. 2.3.

6. Un argument d'un type énuméré serait certainement plus explicite que le booléen utilisé ici.

7. https://fr.wikipedia.org/wiki/Algorithme_de_tri (consulté le 13 septembre 2021).

8. <https://en.cppreference.com/w/c/algorithm/qsort> (consulté le 13 septembre 2021).

9. Par exemple, $(4 \bmod 3) < (1 \bmod 3)$, car $(4 \bmod 3 = 1)$ tandis que $(2 \bmod 3 = 2)$ et $1 < 2$.