



# CPPLI : TD 2 : C : Tableau statique

Nicolas Vansteenkiste      Romain Absil      Jonas Beleho \*  
(ESI – HE2B)

Année académique 2019 – 2020

Ce TD<sup>1</sup> aborde les tableaux statiques du langage C dans sa version C11<sup>2</sup> (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`<sup>3</sup> d'`int` en argument, séparés par un espace. Un passage à la ligne termine l'affichage. Si `data` vaut `NULL`<sup>4</sup> ou si le tableau contient moins de `nbElem` éléments, la fonction adopte un `comportement indéterminé`<sup>5</sup>.

**Ex. 2.2** Écrivez la fonction de prototype :

```
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`<sup>6</sup>. Implémentez l'`algorithme de`

---

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

1. [https://poesi.esi-bru.be/pluginfile.php/1320/mod\\_folder/content/0/td02\\_c/td02\\_c.pdf](https://poesi.esi-bru.be/pluginfile.php/1320/mod_folder/content/0/td02_c/td02_c.pdf)

2. [https://en.wikipedia.org/wiki/C11\\_\(C\\_standard\\_revision\)](https://en.wikipedia.org/wiki/C11_(C_standard_revision))

3. <https://en.cppreference.com/w/c/language/array>

4. <https://en.cppreference.com/w/c/types/NULL>

5. <http://blog.llvm.org/2011/05/what-every-c-programmer-should-know.html>

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

`tri`<sup>7</sup> 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`<sup>8</sup>, triez un tableau d'`int` dans l'ordre :

- (a) croissant ;
- (b) décroissant ;
- (c) croissant 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.

---

7. [https://fr.wikipedia.org/wiki/Algorithme\\_de\\_tri](https://fr.wikipedia.org/wiki/Algorithme_de_tri)

8. <https://en.cppreference.com/w/c/algorithm/qsort>