Exercises — Insertion Sort

version **#**

Assistants C/Unix 2022 <assistants@tickets.assistants.epita.fr>

Copyright

The use of this document must abide by the following rules:

- ▷ You downloaded it from the assistants' intranet.*
- ▷ This document is strictly personal and must **not** be passed onto someone else.
- ▷ Non-compliance with these rules can lead to severe sanctions.

Contents

*https://intra.assistants.epita.fr

1 Insertion Sort

Files to submit:

- insertion_sort/insertion_sort.c

Authorized functions: You are only allowed to use the following functions:

- malloc(3)
- calloc(3)
- free(3)

Authorized headers: You are only allowed to use the functions defined in the following headers:

- errno.h
- err.h
- assert.h
- stddef.h

1.1 Goal

You have to implement a *generic* insertion sort. It will take two arguments: an array of pointers and a **comparison function**. The array of pointers is **NULL** terminated to indicate the end. The function prototype is:

```
typedef int (*f_cmp)(const void*, const void*);
void insertion_sort(void **array, f_cmp comp);
```

The argument `comp` is a pointer to a comparison function which returns an integer lower, equal or greater than zero if the first argument is respectively lower, equal or greater than the second argument. For example, the function `strcmp(3)` matches this description.

Your implementation performance will be tested: don't try to trick us with a simple bubble sort.

It is my job to make sure you do yours.