Exercises — Int Vector Insertion Sort

version **#7be580532266ed398481e31366afcc24b1950c2a**

Assistants C/Unix 2023 <assistants@tickets.assistants.epita.fr>

Copyright

This document is for internal use at EPITA (website) only.

Contents

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

File Tree

```
int_vector_insert_sort/
├── int_vector_insert_sort.c  (to submit)
└── int_vector_insert_sort.h
```

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

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

Compilation : Your code must compile with the following flags

- -std=c99 -pedantic -Werror -Wall -Wextra -Wvla

Main function : None

1 Goal

Write a function that takes a vector of integers and sorts it completely in ascending order. You will be using the sorting algorithm called *Insertion Sort*. The size of the table will always be correct and superior to zero.

Structure of the vector:

```c
struct int_vector
{
    size_t size;
    int data[INT_VECTOR_LENGTH];
};
```

To help you in this endeavor, you will have to first write another function: `swap`.

```c
static struct int_vector swap(struct int_vector vec, size_t i, size_t j)
```

As the name entails, you need to swap the values at index `i` and `j` and returns the vector. Using a temporary value could help greatly.

2 Example Swap

```c
int main(void)
{
    struct int_vector vec = { .size = 5, .data = { 3, 4, 12, 2, 8 } };

    vec = swap(vec, 0, 1);

    /* vec should be 4 3 12 2 8 */

    return 0;
}
```

3 Insertion sort

Once done, you can use the swap function to code the insertion sort.

```c
struct int_vector int_vector_insert_sort(struct int_vector vec);
```

The way is lit. The path is clear. We require only the strength to follow it.