{EPITECH}
LEARN DIFFERENT*

# WORKSHOP ENTRAINEMENT AU STUMPER

## CRYPTIC MESSAGES WITHOUT BUFFERS

# WORKSHOP ENTRAINEMENT AU STUMPER

You are a secret agent working for a highly classified cryptography agency. Your mission is to develop a covert communication system using a C program. However, there's a catch – you can only use the write function for both input and output operations. Your task is to create an algorithm that can encrypt and decrypt sensitive messages without using any dynamic memory allocation.

You will be rated on your ability to write clean, efficient, and well-documented code.

The use of any libC functions is strictly forbidden. (especially `printf` and `malloc`). `write` is the only function you can use to write to the standard output.

## Instructions

1. Implement a function to encrypt a string given as a parameter. It should only be launched only if the "-e" flag is set in the command line compilation.
2. Implement a function to decrypt a string given as a parameter. It should only be launched only if the "-d" flag is set in the command line compilation.
3. If there is no flag, the key is negative or the string is empty, it should return an error.
4. If the string contains non-alpha characters, the program should return an error.
5. The final string should contain only alpha characters.

If any error occurs, the program should return 84. Otherwise, it should return 0 after the execution.

## Expected outputs

```
Terminal
B-INN-000-stumper> ./my_cesar -e "Bonjour mon choux" 34 | cat -e
Kxwsxda vxw lqxdg$
```

```
Terminal
B-INN-000-stumper> ./my_cesar -d "Kxwsxda vxw lqxdg$" 34 | cat -e
Bonjour mon choux$
```

```
Terminal
B-INN-000-stumper> ./my_cesar -e Bonjour -1 | echo $?
84
```

Remember, just like in regular Stumpers, Unit Testing is mandatory. (Remember the criterion workshops)

v 3.1.1