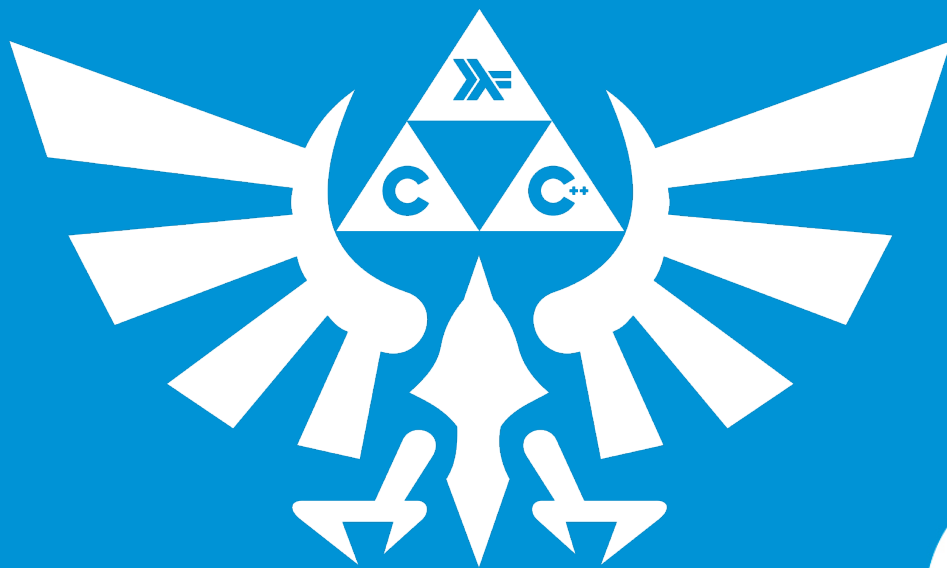




DAY 05

STRING MODULE



DAY 05

All your exercises will be compiled with the `-std=gnu17 -Wall -Wextra` **flags**, unless specified otherwise.

All output goes to the standard output.



None of your files must contain a `main` function, unless specified otherwise. We will use our own `main` functions to compile and test your code.



Your Makefile must build your source files and create a static library called `libstring.a`. You must also submit a file called `string.h` containing the definition of your module (structure and init/destroy functions prototypes).



Read the examples CAREFULLY. They might require things that weren't mentioned in the subject...

Unit Tests

It is highly recommended to test your functions as you implement them. It is common practice to create and use what are called **unit tests**.

From now on, we expect you to write unit tests for your functions (when possible). To do so, please follow the instructions in the **“How to write Unit Tests”** document on the intranet, available [here](#).

Exercise 0 - My String

Create a `string_t` module.

The structure must hold a `char *str` member, and your module should contain an initialization function and a destruction function with the following prototypes :

```
void string_init(string_t *this, const char *s);
void string_destroy(string_t *this);
```

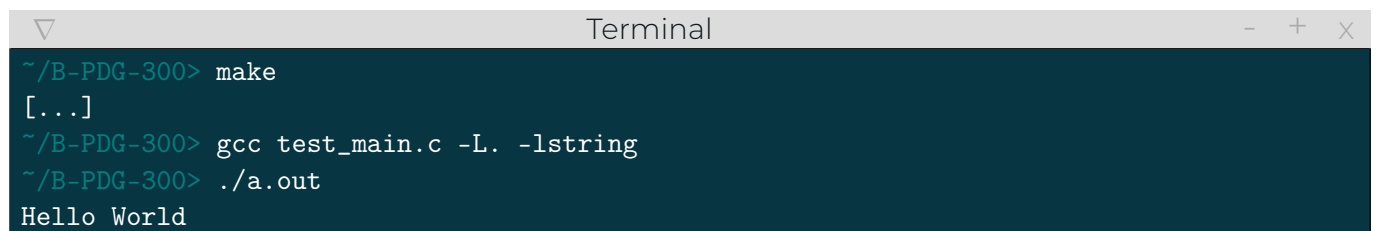
`string_init` assigns the value of `s` to the `str` member of the structure `string_t`.

Here is a sample `main` function and its expected output :

```
#include <stdio.h>
#include "string.h"

int main(void)
{
    string_t s;

    string_init(&s, "Hello World");
    printf("%s\n", s.str);
    string_destroy(&s);
    return (0);
}
```



```
Terminal
~/B-PDG-300> make
[...]
~/B-PDG-300> gcc test_main.c -L. -lstring
~/B-PDG-300> ./a.out
Hello World
```