B1- Unix System Programmation

B-PSU-100

Bootstrap

my_printf

v1.6

EPITECH.
L'ECOLE DE L'INNOVATION ET DE L'EXPERTISE INFORMATIQUE

Bootstrap

my_printf

binary name:	libmy.a
repository name:	PSU_my_printf_bootstrap_$ACADEMICYEAR
repository rights:	ramassage-tek
language:	C
group size:	1
compilation:	via Makefile, including re, clean and fclean rules

- Your repository must contain the totality of your source files, but no useless files (binary, temp files, obj files,...).
- All the bonus files (including a potential specific Makefile) should be in a directory named *bonus*.
- Error messages have to be written on the error output, and the program should then exit with the 84 error code (0 if there is no error).

Everything you need must be found in the delivery directory and its sub-directories (nothing should be found elsewhere, even if it has permissions and absolute paths).

You must write a Makefile that creates a library named *my*.
The **libmy.a** library must contain the **sum_stdarg** and **disp_stdarg** functions (at least), in addition to any other functions required to make it functional.

Here are the prototypes of the functions:

```
int   sum_stdarg(int i, int nb, ...);
int   disp_stdarg(char *s, ...);
```

sum_stdarg returns the sum of the last *nb* arguments if *i*=0.
If *i*=1, it returns the sum of the size of the last *nb* character strings passed as parameter.

Start writing tests for the function before writing the function itself. Think of all the cases (both with i=0 and i=1).

Example:

```
Test(sum_stdarg, return_correct_when_i_is_zero)
{
        int ret = sum_stdarg(0, 3, 21, 25, -4);

        cr_assert_eq(ret, 42);
}
```

disp_stdarg function displays the arguments followed by '\n', in the order in which they were passed, according to the value of *s*, which is composed of the letters *c* (for char), *s* (for char*) and *i* (for int).

Only malloc and free are allowed.

Examples

```
sum_stdarg(0, 3, 1, 2, 3);
```

should return 6

```
sum_stdarg(1, 5, "Hello", "a", "toto", "", "plop");
```

should return 14

```
disp_stdarg("csiis", 'X', "hi", 10, -3, "plop");
```

should display the following:

```
X
hi
10
-3
plop
```