B1 - Unix System Programming

B-PSU-100

my_printf

printf command-like

2.3

my_printf

binary name: libmy.a
language: C
compilation: via Makefile, including re, clean and fclean rules

- The totality of your source files, except all useless files (binary, temp files, obj files,...), must be included in your delivery.
- 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).

You must submit a Makefile that will create a library named my, as well as all source files. The libmy.a library must contain the my_printf function, in addition to any other functions required to make it functional.

You must recode the **printf** function from the C library according to the C99 standard. Your function should be prototyped like the printf function.
You do not have to implement the C library printf buffer handling.
You must process all **printf** formating flags **except** the following (which are optional):

- float or double types management,
- %n flag management,
- " * " , " ' " and " I " (capital i) flags management.

You must add a %b formating flag, which prints unsigned numbers in a binary base.
You must also add a %S formating flag, which prints a character string (like %s). However, non-printable characters (ASCII value strictly smaller than 32 or greater or equal than 127) must be represented by a backslash to be followed by the character's value in octal base.

man 3 printf / man 3 stdarg

The whole libC is forbidden, except **malloc**, **free** and **write**.

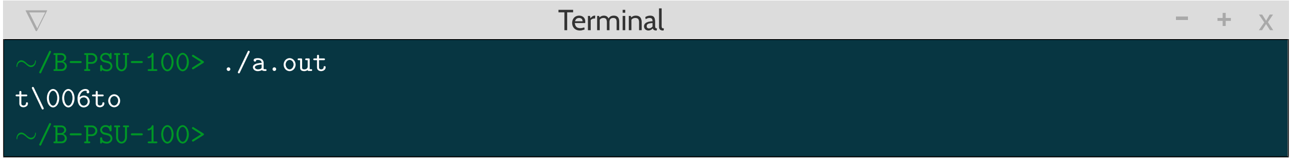
Unit tests

Criterion includes mechanisms to test standard output and standard error, you can learn more about it there…

```c
#include <criterion/criterion.h>
#include <criterion/redirect.h>
#include "my.h"

void redirect_all_std(void)
{
    cr_redirect_stdout();
    cr_redirect_stderr();
}

Test(my_printf, simple_string, .init = redirect_all_std)
{
    my_printf("hello world");
    cr_assert_stdout_eq_str("hello world");
}
```

Examples

```c
char str[5];

my_strcpy(str, "toto");
str[1] = 6;
my_printf("%S\n", str);
```



```
~/B-PSU-100> ./a.out
t\006to
~/B-PSU-100>
```