BB1-

B-PSU-050

my_printf

printf command-like

v1.3

my_printf

printf command-like

binary name:	libmy.a
repository name:	PSU_$YEAR_my_printf
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).

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. You function should be prototyped like the printf function.
You do no have to perform the C library printf buffer handling.
You must process all **printf** formatting flags **except** the following (which are optional):

- float or double types management,
- %n flag management.

You must add %b handling, which prints unsigned numbers in a binary base.
You must also add %S handling, which prints a character string (like %s). However, non-printable characters (ASCII value stricly 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.

Example

```
char str[5];
my_strcpy (str, "toto");
str[1] = 6;
my_printf("%S", str);
```

Terminal + x

```
~/B-PSU-050> ./a.out
t\006to
```