



B1 - Unix System Programming

B-PSU-100

my_printf

Bootstrap



2.0



my_printf

binary name: libmy.a
repository name: PSU_my_printf_bootstrap_\$ACADEMICYEAR
repository rights: ramassage-tek
language: C
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,...).

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, ... );
```

It 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.

```
int disp_stdarg( char *s, ... );
```

This 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, free and write are allowed.



UNIT TESTS



It's generally considered good practice to write its unit tests before starting to implement a function. Think of all the cases!

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

    cr_assert_eq(ret, 42);
}
```

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
```