



# B1- Unix and C Lab Seminar

B-CPE-100

Day 07

Libmy, arguments





# Day 07

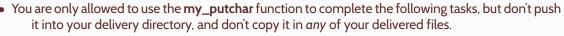
#### Libmy, arguments

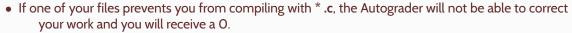
repository name: CPool\_DayO7\_\$ACADEMICYEAR

repository rights: ramassage-tek

language: C group size: 1

- Your repository must contain the totality of your source files, but no useless files (binary, temp files, obj files,...).
- Don't push your main function into your delivery directory, we will be adding our own. Your files will be compiled adding our main.c and our my\_putchar.c files.







All .c files from your delivery folder will be collected and compiled with your libmy, which is found in CPool\_Day07\_\$ACADEMICYEAR/lib/my. For those of you using .h files, they must be located in CPool\_Day07\_\$ACADEMICYEAR/include.

Some tests will automatically compile your functions the following way:

```
Terminal - + x

~/B-CPE-100> cd task01

~/B-CPE-100> gcc *.c -c -I../include/

~/B-CPE-100> gcc *.o ~autograder/main_task01.o -L../lib/my/ -o task01 -lmy
```



Create your repository at the beginning of the day and submit your work on a regular basis! The delivery directory is specified within the instructions for each task. In order to keep your repository clean, pay attention to gitignore.



Don't forget to write unit tests for all your functions! Check out DayO4 if you need an example, and re-read this document.





## Task 01

### libmy.a

Build your own library in  $CPool\_DayO7\_\$ACADEMICYEAR/lib/my$  and name it libmy.a

The library MUST contain ALL of the following functions:

```
void my_putchar(char c);
                                                   16 char *my_strstr(char const *str, char const
   int my_isneg(int nb);
                                                        *to_find);
                                                   17 int my_strcmp(char const *s1, char const *s2);
3
  int my_put_nbr(int nb);
  int my_swap(int *a, int *b);
                                                   18 int my_strncmp(char const *s1, char const *s2,
  int my_putstr(char const *str);
                                                        int n):
   int my_strlen(char const *str);
                                                   19 char *my_strupcase(char *str);
                                                   20 char *my_strlowcase(char *str);
  int my_getnbr(char const *str);
8
  void my_sort_int_array(int *tab, int size);
                                                   21 char *my_strcapitalize(char *str);
  int my_compute_power_rec(int nb, int power);
                                                   22 int my_str_isalpha(char const *str);
                                                   23 int my_str_isnum(char const *str);
10 int my_compute_square_root(int nb);
11 int my_is_prime(int nb);
                                                   24 int my_str_islower(char const *str);
12 int my_find_prime_sup(int nb);
                                                   25 int my_str_isupper(char const *str);
13 char *my_strcpy(char *dest, char const *src);
                                                   26 int my_str_isprintable(char const *str);
14 char *my_strncpy(char *dest, char const *src,
                                                   27 int my_showstr(char const *str);
                                                   28 int my_showmem(char const *str, int size);
                                                   29 char *my_strcat(char *dest, char const *src);
15 char *my_revstr(char *str);
                                                   30 char *my_strncat(char *dest, char const *src,
```

Beware to deliver your **libmy.a** library in the correct folder because it will be used to compile all of your programs. **Delivery:** CPool\_DayO7\_\$ACADEMICYEAR/lib/my/libmy.a



The functions from the following two tasks must be included in your library.

From tomorrow onwards, none of the functions present in your library must be present in your sources.

### Task 02

my\_strcat

Write a function that concatenates two strings. It must be prototyped the following way:

```
char *my_strcat(char *dest, char const *src);
```

Delivery: CPool\_DayO7\_\$ACADEMICYEAR/my\_strcat.c







## Task 03

### my\_strncat

Write a function that concatenates n characters of the **src** string to the end of the **dest** string. It must be prototyped the following way:

```
char *my_strncat(char *dest, char const *src, int nb);
```

**Delivery:** CPool\_DayO7\_\$ACADEMICYEAR/my\_strncat.c

## Task 04

### my\_print\_params

Write a program that displays its arguments (received on the command line). Since it is a PROGRAM, you need to put the **main** function in your delivered file.

You are to display all arguments (including  ${\tt argv}\, {\tt [0]}$ ), on different lines.

**Delivery:** CPool\_DayO7\_\$ACADEMICYEAR/taskO4/my\_print\_params.c



Your main function must return 0.





# Task 05

#### my\_rev\_params

Write a program that displays all the arguments received on the command line in reverse order. You are to display all arguments (including argv [0]), on different lines.

Delivery: CPool\_DayO7\_\$ACADEMICYEAR/taskO5/my\_rev\_params.c



Your main function must return 0.

```
Terminal - + X

~/B-CPE-100> ./a.out test "This is a test " retest | cat -e

retest$

This is a test $

test$
./a.out$
```

## Task 06

#### my\_sort\_params

Write a program that displays all its arguments, in **ascii** order. You are to display all arguments (including argv [0]), on different lines. **Delivery:** CPool\_DayO7\_\$ACADEMICYEAR/taskO6/my\_sort\_params.c



Your main function must return 0.

```
Terminal - + x

~/B-CPE-100> ./a.out test "This is a test " retest | cat -e
./a.out$

This is a test $

retest$

test$
```

