B1- Unix and C Lab Seminar

B-CPE-100

Day 07

Libmy, arguments

v1.12

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

Day 07

Libmy, arguments

repository name:	CPool_Day07_$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**.
- If one of your files prevents you from compiling with * **.c**, the Autograder will not be able to correct your work and you will receive a 0.

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

```
~/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`.

Allowed system function(s): write

Don't forget to write unit tests for all your functions!
Check out Day04 if you need an example, and re-read this document.

Task 01

libmy.a

Build your own library in **CPool_Day07_$ACADEMICYEAR/lib/my** and name it *libmy.a*
The library **MUST** contain **ALL** of the following functions:

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

Beware to deliver your **libmy.a** library in the correct folder because it will be used to compile all of your programs.
Delivery: CPool_Day07_$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.

All the source code used to build the library must be present in your **lib/my** directory on your repository

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_Day07_$ACADEMICYEAR/my_strcat.c

man strcat

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_Day07_$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 `argv[0]`), on different lines.
Delivery: CPool_Day07_$ACADEMICYEAR/task04/*.c

Your main function must return `0`.

Terminal

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

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_Day07_$ACADEMICYEAR/task05/*.c

> Your main function must return `0`.

Terminal

```
~/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_Day07_$ACADEMICYEAR/task06/*.c

> Your main function must return `0`.

Terminal

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