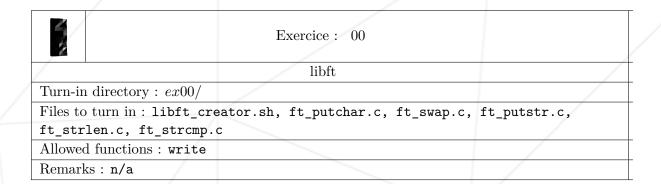
Chapter III

Exercise 00: libft



- Create your ft library. It'll be called libft.a.
- A shell script called libft_creator.sh will compile source files appropriately and will create your library.
- This library should contain <u>all</u> of the following functions :

```
void ft_putchar(char c);
void ft_swap(int *a, int *b);
void ft_putstr(char *str);
int ft_strlen(char *str);
int ft_strcmp(char *s1, char *s2);
```

• We'll launch the following command-line :

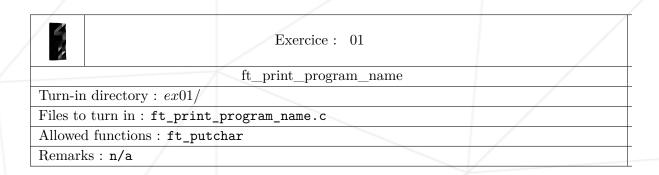
```
sh libft_creator.sh
```



Don't hesitate to add other useful functions... ;-)

Chapter IV

Exercise 01: ft_print_program_name

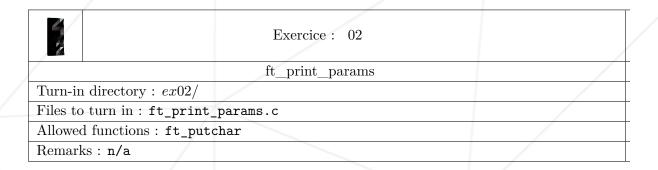


- We're dealing with a <u>program</u> here, you should therefore have a function main in your .c file.
- Create a program that displays its own name.
- Example:

```
$>./a.out
./a.out
```

Chapter V

Exercise 02: ft_print_params



- We're dealing with a <u>program</u> here, you should therefore have a function main in your .c file.
- Create a program that displays its given arguments.
- Example:

```
$>./a.out test1 test2 test3
test1
test2
test3
$>
```

Chapter VI

Exercise 03: ft_rev_params

Exercice: 03	
ft_rev_params	/
arams.c	
char	
	ft_rev_params

- ullet We're dealing with a <u>program</u> here, you should therefore have a function main in your .c file.
- Create a program that displays its given arguments in reverse order.
- It should display all arguments, except for argv[0].
- All arguments have to have their own line.