C Piscine

Day 03

Staff 42 pedago@42.fr

Summary: This document is the subject for Day03 of the C Piscine @ 42.

Contents

1 Instructions 2

2 Topics 4

3 Foreword 5

4 Exercise 00 : ft_ft 6

5 Exercise 01 : ft_ultimate_ft 7

6 Exercise 02 : ft_swap 8

7 Exercise 03 : ft_div_mod 9

8 Exercise 04 : ft_ultimate_div_mod 10

9 Exercise 05 : ft_putstr 11

10 Exercise 06 : ft_strlen 12

11 Exercise 07 : ft_strrev 13

12 Exercise 08 : ft_atoi 14

13 Exercise 09 : ft_sort_integer_table 15

Chapter 1

Instructions

- The exercises are carefully laid out in order of difficulty, from easiest to hardest. An exercise is only graded if all previous ones are correct. In other words: the grading for a day stops at the first mistake.
- Be mindful of the submission procedures indicated at the start of every exercise.
- Your exercises will be checked and graded by your fellow classmates.
- On top of that, your exercises will be checked and graded by a program called Moulinette.
- Moulinette is very meticulous and strict in its evaluation of your work. It is entirely automated and there is no way to negotiate with it. Be as thorough as possible!
- Moulinette relies on a program called **norminette** to check if your files respect the Norm. An exercise containing files that do not respect the Norm will be graded 0.
- Using a forbidden function is considered cheating. Cheaters get -42, and this grade is non-negotiable.
- If **ft_putchar()** is an authorized function, we will compile your code with our **ft_putchar.c**.
- You'll only have to submit a **main()** function if we ask for a program.
- Moulinette compiles with these flags: **-Wall -Wextra -Werror**, and uses **gcc**.
- If your program doesn't compile, it will be graded 0.
- You should not leave any additional file in your directory than those specified in the subject.

`norminette` must be launched with the `-R CheckForbiddenSourceHeader` flag. Moulinette will use it too.

The forewords are entirely unrelated to the subjects and can safely be ignored.

Chapter 2

Topics

Today, you will have to learn about:

- Pointers
- NULL terminated strings

Chapter 3

Foreword

Vincent: And you know what they call a... a... a Quarter Pounder with Cheese in Paris?

Jules: They don't call it a Quarter Pounder with cheese?

Vincent: No man, they got the metric system. They wouldn't know what the fuck a Quarter Pounder is.

Jules: Then what do they call it?

Vincent: They call it a Royale with cheese.

Jules: A Royale with cheese. What do they call a Big Mac?

Vincent: Well, a Big Mac's a Big Mac, but they call it le Big-Mac.

Jules: Le Big-Mac. Ha ha ha ha. What do they call a Whopper?

Vincent: I dunno, I didn't go into Burger King.

At least one of the following exercices has nothing to do with a Royale with cheese.

Chapter 4

Exercise 00 : ft_ft

Turn-in directory : ex00/
Files to turn in: ft_ft.c
Allowed functions: None

- Create a function that takes a pointer to an int as a parameter, and stores the value 42 in that int.
- Here's how it should be prototyped :

```
void         ft_ft(int *nbr);
```

Chapter 5

Exercise 01 : ft_ultimate_ft

Turn-in directory : ex01/
Files to turn in: ft_ultimate_ft.c
Allowed functions: None

- Create a function that takes a pointer to pointer to pointer to pointer to pointer to pointer to pointer to pointer to pointer to an int as a parameter and stores the value 42 in that int.
- Here's how it should be prototyped :

```
void        ft_ultimate_ft(int *********nbr);
```

Chapter 6

Exercise 02 : ft_swap

Turn-in directory : ex02/

Files to turn in: ft_swap.c

Allowed functions: None

- Create a function that swaps the value of two integers whose addresses are passed as parameters.
- Here's how it should be prototyped :

```
void     ft_swap(int *a, int *b);
```

Chapter 7

Exercise 03 : ft_div_mod

Turn-in directory : ex03/
Files to turn in: ft_div_mod.c
Allowed functions: None

- Create a function **ft_div_mod** prototyped like this :

```
void    ft_div_mod(int a, int b, int *div, int *mod);
```

- This function divides parameters **a** by **b** and stores the result in the int pointed by **div**. It also stores the remainder of the division of **a** by **b** in the int pointed by **mod**.

Chapter 8

Exercise 04 : ft_ultimate_div_mod

Turn-in directory : ex04/

Files to turn in: ft_ultimate_div_mod.c

Allowed functions: None

- Create a function **ft_ultimate_div_mod** with the following prototype :

```
void    ft_ultimate_div_mod(int *a, int *b);
```

- This function divides parameters **a** by **b**. The result of this division is stored in the int pointed by **a**. The remainder of the division is stored in the int pointed by **b**.

Chapter 9

Exercise 05 : ft_putstr

Turn-in directory : ex05/
Files to turn in: ft_putstr.c
Allowed functions: ft_putchar

- Create a function that displays a string of characters on the standard output.
- Here's how it should be prototyped :

```
        void     ft_putstr(char *str);
```

Chapter 10

Exercise 06 : ft_strlen

Turn-in directory : ex06/

Files to turn in: ft_strlen.c

Allowed functions: None

- Create a function that counts and returns the number of characters in a string.
- Here's how it should be prototyped :

```
int         ft_strlen(char *str);
```

Chapter 11

Exercise 07 : ft_strrev

Turn-in directory : ex07/

Files to turn in: ft_strrev.c

Allowed functions: None

- Create a function that reverses the order of characters in a string.
- It has to return str.
- Here's how it should be prototyped :

```c
char        *ft_strrev(char *str);
```

- Example:

```
a => a
ab => ba
abcde => edcba
```

Chapter 12

Exercise 08 : ft_atoi

Turn-in directory : ex08/

Files to turn in: ft_atoi.c

Allowed functions: None

- Reproduce the behavior of the function **atoi** (man atoi).
- Here's how it should be prototyped :

```
int     ft_atoi(char *str);
```

Chapter 13

Exercise 09 : ft_sort_integer_table

Turn-in directory : ex09/
Files to turn in: ft_sort_integer_table.c
Allowed functions: None

- Create a function which sorts an array of integers by ascending order.
- The arguments are a pointer to int and the number of ints in the array.
- Here's how it should be prototyped :

```
void    ft_sort_integer_table(int *tab, int size);
```