# C Piscine

Day 02

Staff 42 pedago@42.fr

*Summary: This document is the subject for Day02 of the C Piscine @ 42.*

# Contents

# 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 start programming in C. You will have to learn about, among other things:

- Using the C compiler
- Printing to the standard output
- Functions
- Loops

# Chapter 3

## Foreword

**Cod liver oil is a nutritional supplement derived from liver of cod fish (Gadidae).**

**As with most fish oils, it has high levels of the omega-3 fatty acids, eicosapentaenoic acid (EPA) and docosahexaenoic acid (DHA).**
**Cod liver oil also contains vitamin A and vitamin D.**

**It has historically been taken because of its vitamin A and vitamin D content.**

**It was once commonly given to children, because vitamin D has been shown to prevent rickets and other symptoms of vitamin D deficiency.**

Contrary to Cod liver oil, C is good, eat some!

# Chapter 4

## Exercise 00 : ft_print_alphabet

**Turn-in directory :** `ex00/`

**Files to turn in:** `ft_print_alphabet.c`

**Allowed functions:** `ft_putchar`

---

- Create a function that displays the alphabet in lowercase, on a single line, by ascending order, starting from the letter 'a'.
- Here's how it should be prototyped :

```
void ft_print_alphabet(void);
```

# Chapter 5

## Exercise 01 : ft_print_reverse_alphabet

**Turn-in directory :** `ex01/`

**Files to turn in:** `ft_print_reverse_alphabet.c`

**Allowed functions:** `ft_putchar`

---

- Create a function that displays the alphabet in lowercase, on a single line, by descending order, starting from the letter ‘z’.
- Here’s how it should be prototyped :

```
void ft_print_reverse_alphabet(void);
```

# Chapter 6

## Exercise 02 : ft_print_numbers

**Turn-in directory :** `ex02/`
**Files to turn in:** `ft_print_numbers.c`
**Allowed functions:** `ft_putchar`

---

- Create a function that displays all digits, on a single line, by ascending order.
- Here's how it should be prototyped :

```
void ft_print_numbers(void);
```

# Chapter 7

## Exercise 03: ft_is_negative

**Turn-in directory :** `ex03/`
**Files to turn in:** `ft_is_negative.c`
**Allowed functions:** `ft_putchar`

---

- Create a function that displays 'N' or 'P' depending on the integer's sign entered as a parameter. If **n** is negative, display 'N'. If **n** is positive or null, display 'P'.
- Here's how it should be prototyped :

```
void ft_is_negative(int n);
```

# Chapter 8

## Exercise 04 : ft_print_comb

**Turn-in directory :** `ex04/`
**Files to turn in:** `ft_print_comb.c`
**Allowed functions:** `ft_putchar`

---

- Create a function that displays all different combinations of three different digits in ascending order, listed by ascending order - yes, repetition is voluntary.
- Here's the intended output :

```
$>./a.out | cat -e
012, 013, 014, 015, 016, 017, 018, 019, 023, ..., 789$>
```

- 987 isn't there because 789 already is.
- 999 isn't there because the digit 9 is present more than once.
- Here's how it should be prototyped :

```
void ft_print_comb(void);
```

# Chapter 9

## Exercise 05 : ft_print_comb2

**Turn-in directory :** `ex05/`
**Files to turn in:** `ft_print_comb2.c`
**Allowed functions:** `ft_putchar`

---

- Create a function that displays all different combination of two digits between 00 and 99, listed by ascending order.
- Here's the expected output :

```
$>./a.out | cat -e
00 01, 00 02, 00 03, 00 04, 00 05, ..., 00 99, 01 02, ..., 97 99, 98 99$>
```

- Here's how it should be prototyped :

```
void ft_print_comb2(void);
```

# Chapter 10

## Exercise 06 : ft_putnbr

**Turn-in directory :** `ex06/`
**Files to turn in:** `ft_putnbr.c`
**Allowed functions:** `ft_putchar`

---

- Create a function that displays the number entered as a parameter. The function has to be able to display all possible values within an **int** type variable.
- Here's how it should be prototyped :

```
void ft_putnbr(int nb);
```

- For example:
    - **ft_putnbr(42)** displays "**42**".

# Chapter 11

## Exercise 07 : ft_print_combn

**Turn-in directory :** ex07/
**Files to turn in:** ft_print_combn.c
**Allowed functions:** ft_putchar

---

- Create a function that displays all different combinations of **n** numbers by ascending order.
- n will be so that : 0 < n < 10.
- If **n** = 2, here's the expected output :

```
$>./a.out | cat -e
01, 02, 03, ..., 09, 12, ..., 79, 89$>
```

- Here's how it should be prototyped :

```
void ft_print_combn(int n);
```