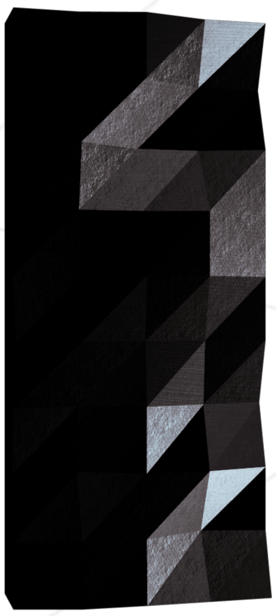

C Piscine

C 00

Summary: This document serves as the subject for the C 00 module of the C Piscine at 42.

Version: 7.8

Contents

Chapter I

Instructions

- Only this page serves as your reference, do not trust rumors.
- Watch out! This document may change before submission.
- Ensure you have the appropriate permissions on your files and directories.
- You must follow the **submission procedures** for all your exercises.
- Your exercises will be checked and graded by your fellow classmates.
- Additionally, your exercises will be evaluated by a program called **Moulinette**.
- **Moulinette** is meticulous and strict in its assessment. It is fully automated, and there is no way to negotiate with it. To avoid unpleasant surprises, be as thorough as possible.
- **Moulinette** is not open-minded. If your code does not adhere to the Norm, it won't attempt to understand it. **Moulinette** relies on a program called **norminette** to check if your files comply with the Norm. TL;DR: Submitting work that doesn't pass **norminette**'s check makes no sense.
- These exercises are arranged in order of difficulty, from easiest to hardest. We **will not** consider a successfully completed harder exercise if an easier one is not fully functional.
- Using a forbidden function is considered cheating. Cheaters receive a grade of **-42**, which is non-negotiable.
- You only need to submit a **main()** function if we specifically ask for a **program**
- **Moulinette** compiles with the following flags: **-Wall -Wextra -Werror**, using **cc**.
- If your program does not compile, you will receive a grade of **0**.
- You **cannot** leave **any** additional file in your directory beyond those specified in the assignment.
- Have a question? Ask the peer on your right. If not, try the peer on your left.

- Your reference guide is called **Google / man / the Internet / ...**
- Check the "C Piscine" section of the forum on the intranet or the Piscine on Slack.
- Carefully examine the examples. They may contain crucial details that are not explicitly stated in the assignment...
- By Odin, by Thor! Use your brain!!!

git branch

`Don't forget to include the` *`standard 42 header`* `in each of your .c and .h files. Norminette checks for its presence anyway!`

`Norminette must be run with the` *`-R CheckForbiddenSourceHeader`* `flag. Moulinette will use it too.`

Chapter II

Foreword

Cod liver oil is a nutritional supplement derived from the liver of codfish (Gadidae).

Like most fish oils, it contains high levels of omega-3 fatty acids, including eicosapentaenoic acid (EPA) and docosahexaenoic acid (DHA). Cod liver oil is also rich in vitamins A and D.

Historically, it has been consumed for its vitamin A and vitamin D content.

It was once commonly given to children, as 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 III

Exercice 00: ft_putchar

	Exercise 00
ft_putchar	
Turn-in directory : *ex*00/	
Files to turn in : `ft_putchar.c`	
Allowed functions : `write`	

- Write a function that displays the character passed as a parameter.
- The function should be prototyped as follows:

```
void ft_putchar(char c);
```

- To display the character, you must use the **write** function as follows:

```
write(1, &c, 1);
```

The first retry delay is short, so don't hesitate to trigger an intermediate evaluation to track your progress.

Chapter IV

Exercise 01: ft_print_alphabet

	Exercise 01
ft_print_alphabet	
Turn-in directory : *ex*01/	
Files to turn in : `ft_print_alphabet.c`	
Allowed functions : `write`	

- Create a function that displays the alphabet in lowercase, on a single line, in ascending order, starting from the letter 'a'.
- The function should be prototyped as follows:

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

Don't hesitate to randomly ask someone in your cluster if you have a question.

Chapter V

Exercise 02: ft_print_reverse_alphabet

	Exercise 02
ft_print_reverse_alphabet	
Turn-in directory : *ex*02/	
Files to turn in : `ft_print_reverse_alphabet.c`	
Allowed functions : `write`	

- Create a function that displays the alphabet in lowercase, on a single line, in descending order, starting from the letter 'z'.
- The function should be prototyped as follows:

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

Push your code to Git regularly!

Chapter VI

Exercise 03: ft_print_numbers

	Exercise 03
ft_print_numbers	
Turn-in directory : *ex*03/	
Files to turn in : `ft_print_numbers.c`	
Allowed functions : `write`	

- Create a function that displays all digits on a single line, in ascending order.
- The function should be prototyped as follows:

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

`Collaboration is the key to success.`

Chapter VII

Exercise 04: ft_is_negative

	Exercise 04
ft_is_negative	
Turn-in directory : *ex*04/	
Files to turn in : `ft_is_negative.c`	
Allowed functions : `write`	

- Create a function that displays 'N' or 'P' depending on the sign of the integer passed as a parameter.
  - If **n** is negative, display 'N'.
  - If **n** is positive or zero, display 'P'.
- The function should be prototyped as follows:

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

`Failure is part of your learning journey.`

Milestone Achieved, Keep Going!

You have reached the end of the mandatory exercises required to validate this project.

Now, it's up to you to decide whether to continue with the following optional exercises or move on to your next project. Both paths will expose you to valuable concepts sooner or later.

To make your decision, consider the following points:

- The very first exam focuses on C programming. If you've already worked on the first C project, this experience will be useful. The same applies to the "rush" at the end of the week (you'll soon learn more about it).
- Your excellence in this Piscine is evaluated based on multiple factors. Completing each project is important, but your overall progress across all Piscine projects also plays a key role. Choose wisely to optimize your results.
- You can always revisit and retry the same project in a few days or weeks, up until the end of the Piscine.
- Staying synchronized with your peers fosters better collaboration.

Chapter VIII

Exercise 05: ft_print_comb

	Exercise 05
ft_print_comb	
Turn-in directory : *ex*05/	
Files to turn in : `ft_print_comb.c`	
Allowed functions : `write`	

- Create a function that displays all different combinations of three distinct digits in ascending order, listed in ascending order (yes!, the repetition is intentional).
- Expected output:

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

- **987** is not included because **789** already covers that combination.
- **999** is not included because the digit **9** appears more than once.
- The function should be prototyped as follows:

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

Did you check with your neighbor on the right?

Chapter IX

Exercise 06: ft_print_comb2

	Exercise 06
ft_print_comb2	
Turn-in directory : *ex*06/	
Files to turn in : `ft_print_comb2.c`	
Allowed functions : `write`	

- Create a function that displays all different combinations of two two-digits numbers (XX XX) between **00** and **99**, listed in ascending order.
- Expected output:

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

- The function should be prototyped as follows:

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

Get inspired by others, but don't let them do the work for you!

Chapter X

Exercise 07: ft_putnbr

	Exercise 07
ft_putnbr	
Turn-in directory : *ex*07/	
Files to turn in : `ft_putnbr.c`	
Allowed functions : `write`	

- Create a function that displays the number passed as a parameter. The function must be able to display all possible values of an **int** type variable.
- The function should be prototyped as follows:

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

- Example:
  - **ft_putnbr(42);** should display **42**.

`Don't trust any single source of information, always run your own tests, checks, and verifications!`

Chapter XI

Exercise 08: ft_print_combn

	Exercise 08
ft_print_combn	
Turn-in directory : *ex*08/	
Files to turn in : `ft_print_combn.c`	
Allowed functions : `write`	

- Create a function that displays all different combinations of **n** digits in ascending order.
- The value of **n** will be such that: $0 < \mathbf{n} < 10$.
- Example output for $\mathbf{n} = 2$:

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

- The function should be prototyped as follows:

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

Did you check with your neighbor on the left?

Chapter XII

Submission and peer-evaluation

Submit your assignment in your **Git** repository as usual. Only the work inside your repository will be evaluated during the defense. Make sure to double-check the names of your files to ensure they are correct.

⚠ `You must submit only the files specified in the project instructions.`