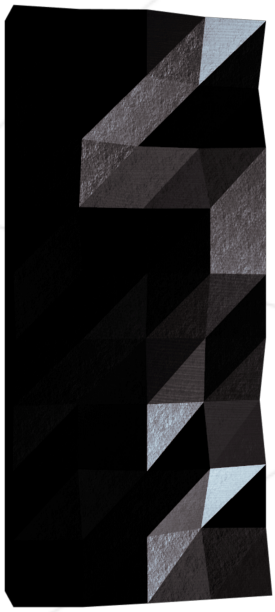

C Piscine

C 01

Summary: This document is the subject of Module C 01 of the C Piscine at 42

Version: 6

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!!!

Do not forget to add the *standard 42 header* in each of your .c/.h files. The norminette check its existence anyway!

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

Chapter II

AI Instructions

Context

The C Piscine is intense. It's your first big challenge at 42 — a deep dive into problem-solving, autonomy, and community.

During this phase, your main objective is to build your foundation — through struggle, repetition, and especially **peer-learning** exchange.

In the AI era, shortcuts are easy to find. However, it's important to consider whether your AI usage is truly helping you grow — or simply getting in the way of developing real skills.

The Piscine is also a human experience — and for now, nothing can replace that. Not even AI.

For a more complete overview of our stance on AI — as a learning tool, as part of the ICT curriculum, and as a growing expectation in the job market — please refer to the dedicated FAQ available on the intranet.

Main message

- Build strong foundations without shortcuts.
- Really develop tech & power skills.
- Experience real peer-learning, start learning how to learn and solve new problems.
- The learning journey is more important than the result.
- Learn about the risks associated with AI, and develop effective control practices and countermeasures to avoid common pitfalls.

Learner rules:

- You should apply reasoning to your assigned tasks, especially before turning to AI.
- You should not ask for direct answers to the AI.
- You should learn about 42 global approach on AI.

Phase outcomes:

Within this foundational phase, you will get the following outcomes:

- Get proper tech and coding foundations.
- Know why and how AI can be dangerous during this phase.

Comments and example:

- Yes, we know AI exists — and yes, it can solve your projects. But you're here to learn, not to prove that AI has learned. Don't waste your time (or ours) just to demonstrate that AI can solve the given problem.
- Learning at 42 isn't about knowing the answer — it's about developing the ability to find one. AI gives you the answer directly, but that prevents you from building your own reasoning. And reasoning takes time, effort, and involves failure. The path to success is not supposed to be easy.
- Keep in mind that during exams, AI is not available — no internet, no smartphones, etc. You'll quickly realise if you've relied too heavily on AI in your learning process.
- Peer learning exposes you to different ideas and approaches, improving your interpersonal skills and your ability to think divergently. That's far more valuable than just chatting with a bot. So don't be shy — talk, ask questions, and learn together!
- Yes, AI will be part of the curriculum — both as a learning tool and as a topic in itself. You'll even have the chance to build your own AI software. In order to learn more about our crescendo approach you'll go through in the documentation available on the intranet.

✓ Good practice:

> I'm stuck on a new concept. I ask someone nearby how they approached it. We talk for 10 minutes — and suddenly it clicks. I get it.

✗ Bad practice:

> I secretly use AI, copy some code that looks right. During peer evaluation, I can't explain anything. I fail. During the exam — no AI — I'm stuck again. I fail.

Chapter III

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 exercises has nothing to do with a Royale with cheese.

Today's threshold

The validation threshold for this project is 50%.

It is up to you to determine which exercises allow you to reach this threshold, and whether you want to complete more.

Chapter IV

Exercise 00 : ft_ft

	Exercise 00
ft_ft	
Turn-in directory: *ex*00/	
Files to turn in: `ft_ft.c`	
Allowed functions: `None`	

- Create a function that takes a pointer to an int as a parameter and sets the value of that int to "42".
- The function should be prototyped as follows:

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

Chapter V

Exercise 01 : ft_ultimate_ft

	Exercise 01
ft_ultimate_ft	
Turn-in directory: *ex*01/	
Files to turn in: `ft_ultimate_ft.c`	
Allowed functions: `None`	

- Create a function that takes a pointer to a pointer to a pointer to a pointer to a pointer to a pointer to a pointer to a pointer to a pointer to an int as a parameter and sets the value of that int to “42”.
- The function should be prototyped as follows:

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

Chapter VI

Exercise 02 : ft_swap

	Exercise 02
ft_swap	
Turn-in directory: $ex02/$	
Files to turn in: `ft_swap.c`	
Allowed functions: `None`	

- Create a function that swaps the values of two integers using their addresses received as parameters.
- The function should be prototyped as follows:

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

Chapter VII

Exercise 03 : ft_div_mod

	Exercise 03
ft_div_mod	
Turn-in directory: *ex*03/	
Files to turn in: `ft_div_mod.c`	
Allowed functions: `None`	

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

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

- This function divides '`a`' by '`b`' and stores the result in the integer pointed to by '`div`'. It also stores the remainder of the division of '`a`' by '`b`' in the integer pointed to by '`mod`'.

Chapter VIII

Exercise 04 : ft_ultimate_div_mod

	Exercise 04
ft_ultimate_div_mod	
Turn-in directory: *ex*04/	
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 the value pointed to by '`a`' by the value pointed to by '`b`'. The result of the division is stored in the integer pointed to by '`a`', while the remainder is stored in the integer pointed to by '`b`'.

Chapter IX

Exercise 05 : ft_putstr

	Exercise 05
ft_putstr	
Turn-in directory: $ex05/$	
Files to turn in: `ft_putstr.c`	
Allowed functions: `write`	

- Create a function that displays a string of characters on the standard output.
- The function should be prototyped as follows:

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

Chapter X

Exercise 06 : ft_strlen

	Exercise 06
ft_strlen	
Turn-in directory: *ex*06/	
Files to turn in: `ft_strlen.c`	
Allowed functions: `None`	

- Create a function that counts and returns the number of characters in a string.
- The function should be prototyped as follows:

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

Chapter XI

Exercise 07 : ft_rev_int_tab

	Exercise 07
ft_rev_int_tab	
Turn-in directory: *ex*07/	
Files to turn in: `ft_rev_int_tab.c`	
Allowed functions: `None`	

- Create a function that reverses a given array of integers (the first element becomes the last, and so on).
- The function takes two arguments: a pointer to an int and the number of elements in the array.
- The function should be prototyped as follows:

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

Chapter XII

Exercise 08 : ft_sort_int_tab

	Exercise 08
ft_sort_int_tab	
Turn-in directory: *ex*08/	
Files to turn in: `ft_sort_int_tab.c`	
Allowed functions: `None`	

- Create a function that sorts an array of integers in ascending order.
- The function takes two arguments: a pointer to an int and the number of elements in the array.
- The function should be prototyped as follows:

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

Chapter XIII

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 need to return only the files requested by the subject of this project.`