



C Piscine

C 07

Staff 42 [pedago@42.fr](mailto:pedago@42.fr)

*Summary: This document is the subject for the module C 07 of the C Piscine @ 42.*

*Version: 5*

# Contents

<b>I</b>	<b>Instructions</b>	<b>2</b>
<b>II</b>	<b>Foreword</b>	<b>4</b>
<b>III</b>	<b>Exercise 00 : ft_strdup</b>	<b>6</b>
<b>IV</b>	<b>Exercise 01 : ft_range</b>	<b>7</b>
<b>V</b>	<b>Exercise 02 : ft_ultimate_range</b>	<b>8</b>
<b>VI</b>	<b>Exercise 03 : ft_strjoin</b>	<b>9</b>
<b>VII</b>	<b>Exercise 04 : ft_convert_base</b>	<b>10</b>
<b>VIII</b>	<b>Exercise 05 : ft_split</b>	<b>11</b>
<b>IX</b>	<b>Submission and peer-evaluation</b>	<b>12</b>

# 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!!!



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

# Chapter II

## Foreword

Morty: Rick!

Rick: Uhp-uhp-uhp! Morty, keep your hands off your ding-dong! It's the only way we can speak freely. Look around you, Morty. Do you really think this wuh-world is real? You'd have to be an idiot not to notice all the sloppy details. Look, that guy's putting a bun between two hot dogs.

Morty: I dunno, Rick, I mean, I've seen people do that before.

Rick: Well, look at that old lady. She's-she's walking a cat on a leash.

Morty: Uh, Mrs. Spencer does that all the time, Rick.

Rick: Look, I-I-I don't want to hear about Mrs. Spencer, Morty! She's an idiot! All right, all right, there. Wh-what about that, Morty?

Morty: Okay, okay, you got me on that one.

Rick: Oh, really, Morty? Are you sure you haven't seen that somewhere in real life before?

Morty: No, no, I haven't seen that. I mean, why would a Pop-Tart want to live inside a toaster, Rick? I mean, th-that would be like the scariest place for them to live. Y'know what I mean?

Rick: You're missing the point, Morty. Why would he drive a smaller toaster with wheels? I mean, does your car look like a smaller version of your house? No.

Morty: So, why are they doing this? W-what do they want?

Rick: Well, that would be obvious to you, Morty, if you'd been paying attention. [an ambulance drives past Rick and Morty and stops; open back doors]

Paramedic: We got the President of the United States in here! We need 10cc of concentrated dark matter, stat, or he'll die!

Morty: Concentrated dark matter? They were asking about that in class.

Rick: Yeah, it's a special fuel I invented to travel through space faster than anybody else. These Zigerions are always trying to scam me out of my secrets, but they made a big mistake this time, Morty. They dragged you into this. Now they're gonna pay!


Morty: What do you- w-w-what are we gonna do?

Rick: We're gonna scam the scammers, Morty. And we're gonna take 'em for everything they've got.

The following exercises will be easier to complete if you are a fan of "Rick and Morty"

# Chapter III

## Exercise 00 : ft\_strdup


	Exercise 00
ft_strdup	
Turn-in directory : <i>ex00/</i>	
Files to turn in : <b>ft_strdup.c</b>	
Allowed functions : <b>malloc</b>	

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

```
char *ft_strdup(char *src);
```

# Chapter IV

## Exercise 01 : ft\_range

	Exercise 01
ft_range	
Turn-in directory : <i>ex01/</i>	
Files to turn in : <b>ft_range.c</b>	
Allowed functions : <b>malloc</b>	

- Create a function **ft\_range** which returns an array of **ints**. This **int** array should contain all values between **min** and **max**.
- **Min** included - **max** excluded.
- Here is how it should be prototyped :


```
int *ft_range(int min, int max);
```

- If **min**'value is greater or equal to **max**'s value, a null pointer should be returned.



# Chapter V

## Exercise 02 : ft\_ultimate\_range

	Exercise 02
ft_ultimate_range	
Turn-in directory : <i>ex02/</i>	
Files to turn in : <code>ft_ultimate_range.c</code>	
Allowed functions : <code>malloc</code>	


- Create a function `ft_ultimate_range` which allocates and assigns an array of `ints`. This `int` array should contain all values between `min` and `max`.
- `Min` included - `max` excluded.
- Here is how it should be prototyped :

```
int ft_ultimate_range(int **range, int min, int max);
```

- The size of `range` should be returned (or -1 on error).
- If the value of `min` is greater or equal to `max`'s value, `range` will point to `NULL` and it should return 0.

# Chapter VI

## Exercice 03 : ft\_strjoin


	Exercice 03
ft_strjoin	
Turn-in directory : <i>ex03/</i>	
Files to turn in : <b>ft_strjoin.c</b>	
Allowed functions : <b>malloc</b>	

- Write a function that concatenates all the strings pointed to by **strs**, separated by **sep**.
- **size** is the number of strings in **strs**.
- If **size** is 0, you must return an empty string that can be freed using **free()**.
- Here is how it should be prototyped:

```
char *ft_strjoin(int size, char **strs, char *sep);
```

# Chapter VII

## Exercise 04 : ft\_convert\_base


	Exercise 04
	ft_convert_base
	Turn-in directory : <i>ex04/</i>
	Files to turn in : <b>ft_convert_base.c</b> , <b>ft_convert_base2.c</b>
	Allowed functions : <b>malloc</b> , <b>free</b>

- Create a function that returns the result of the conversion of the string **nbr** from a base **base\_from** to a base **base\_to**.
- **nbr**, **base\_from**, and **base\_to** may be read-only.
- **nbr** will follow the same rules as **ft\_atoi\_base** (from another module). Beware of the characters '+', '-', and whitespace.
- The number represented by **nbr** must fit inside an **int**.
- If a base is incorrect, **NULL** should be returned.
- The returned number must be prefixed by a single, unique '-' if necessary; no whitespace, no '+'.
- Here's how it should be prototyped:

```
char *ft_convert_base(char *nbr, char *base_from, char *base_to);
```

# Chapter VIII

## Exercise 05 : ft\_split

	Exercise 05
ft_split	
Turn-in directory : <i>ex05/</i>	
Files to turn in : <b>ft_split.c</b>	
Allowed functions : <b>malloc</b>	

- Create a function that splits a string of characters based on an additional string of characters.
- You'll have to use each character from the string **charset** as a separator.
- The function should return an array where each element of the array contains the address of a string, wrapped between two separators. The last element of the array should be NULL to indicate the end of the array.
- There cannot be any empty strings in your array. Draw your own conclusions accordingly.
- The string given as an argument won't be modifiable.
- Here's how it should be prototyped:

```
char **ft_split(char *str, char *charset);
```

# Chapter IX

## Submission and peer-evaluation

Submit your assignment to your `Git` repository as usual. Only the work inside your repository will be evaluated during the defense. Make sure to double-check the filenames to ensure they are correct.



You must submit only the files required by the project instructions.