

C Piscine C 04

Staff 42 pedago@42.fr

Summary: This document is the subject for the unit C 04 of the C Piscine @ 42.

# Contents

Ι	Instructions	2
II	Foreword	4
III	Exercise 00 : ft_strlen	6
IV	Exercise 01 : ft_putstr	7
V	Exercise 02 : ft_putnbr	8
VI	Exercise 03 : ft_atoi	9
VII	Exercise 04 : ft_putnbr_base	10
VIII	Exercise 05 : ft_atoi_base	12

### Chapter I

#### Instructions

- Only this page will serve as reference: do not trust rumors.
- Watch out! This document could potentially change before submission.
- Make sure you have the appropriate permissions on your files and directories.
- You have to follow the submission procedures for all your exercises.
- 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. So if you want to avoid bad surprises, be as thorough as possible.
- Moulinette is not very open-minded. It won't try and understand your code if it doesn't respect the Norm. Moulinette relies on a program called norminette to check if your files respect the norm. TL;DR: it would be idiotic to submit a piece of work that doesn't pass norminette's check.
- These exercises are carefully laid out by order of difficulty from easiest to hardest. We will not take into account a successfully completed harder exercise if an easier one is not perfectly functional.
- Using a forbidden function is considered cheating. Cheaters get -42, and this grade is non-negotiable.
- 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, you'll get 0.
- You <u>cannot</u> leave <u>any</u> additional file in your directory than those specified in the subject.
- Got a question? Ask your peer on the right. Otherwise, try your peer on the left.

- $\bullet$  Your reference guide is called Google / man / the Internet / ....
- Check out the "C Piscine" part of the forum on the intranet, or the slack Piscine.
- Examine the examples thoroughly. They could very well call for details that are not explicitly mentioned in the subject...
- By Odin, by Thor! Use your brain!!!



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

#### Chapter II

#### Foreword

Here are the lyrics for City Hunter's theme song "Moonlight Shadow":

The last time ever she saw him
Carried away by a moonlight shadow
He passed on worried and warning
Carried away by a moonlight shadow.
Lost in a riddle that Saturday night
Far away on the other side.
He was caught in the middle of a desperate fight
And she couldn't find how to push through

The trees that whisper in the evening Carried away by a moonlight shadow Sing a song of sorrow and grieving Carried away by a moonlight shadow All she saw was a silhouette of a gun Far away on the other side.

He was shot six times by a man on the run And she couldn't find how to push through

[Chorus]
I stay, I pray

See you in Heaven far away...

I stay, I pray

See you in Heaven one day.

Four A.M. in the morning
Carried away by a moonlight shadow
I watched your vision forming
Carried away by a moonlight shadow
A star was glowing in the silvery night
Far away on the other side
Will you come to talk to me this night
But she couldn't find how to push through

[Chorus]

C Piscine

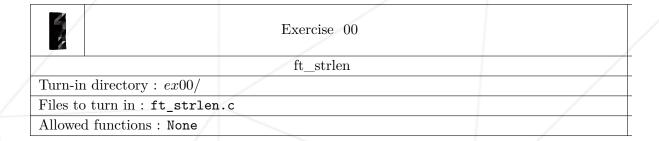
Far away on the other side.

Caught in the middle of a hundred and five
The night was heavy and the air was alive
But she couldn't find how to push through
Carried away by a moonlight shadow
Carried away by a moonlight shadow
Far away on the other side.

Unfortunately, this topic has nothing to do with City Hunter.

# Chapter III

Exercise 00 : ft\_strlen

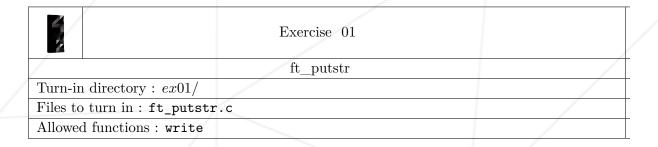


- Create a function that counts and returns the number of characters in a string.
- Here's how it should be prototyped :

int ft\_strlen(char \*str);

# Chapter IV

Exercise 01: ft\_putstr

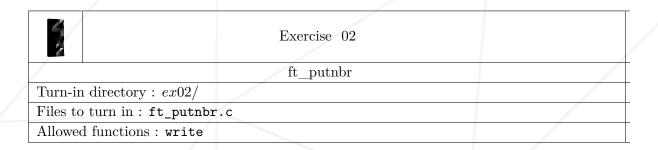


- Create a function that displays a string of characters on the standard output.
- Here's how it should be prototyped :

void ft\_putstr(char \*str);

## Chapter V

# Exercise 02: ft\_putnbr



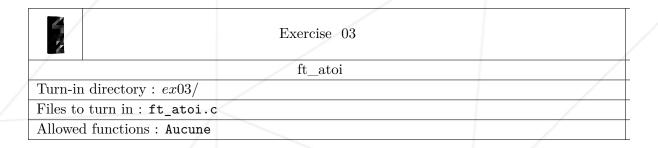
- Create a function that displays the number entered as a parameter. The function has to be able to display all possible values within an int type variable.
- Here's how it should be prototyped:

void ft\_putnbr(int nb);

- For example:
  - o ft\_putnbr(42) displays "42".

## Chapter VI

Exercise 03: ft\_atoi



- Write a function that converts the initial portion of the string pointed by str to its int representation
- The string can start with an arbitray amount of white space (as determined by isspace(3))
- $\bullet$  The string can be followed by an arbitrary amount of + and signs, sign will change the sign of the int returned based on the number of is odd or even.
- Finally the string can be followed by any numbers of the base 10.
- Your function should read the string until the string stop following the rules and return the number found until now.
- You should not take care of overflow or underflow. result can be undefined in that case.
- Here's an example of a program that prints the atoi return value:

```
$>./a.out " ---+--+1234ab567"
-1234
```

• Here's how it should be prototyped:

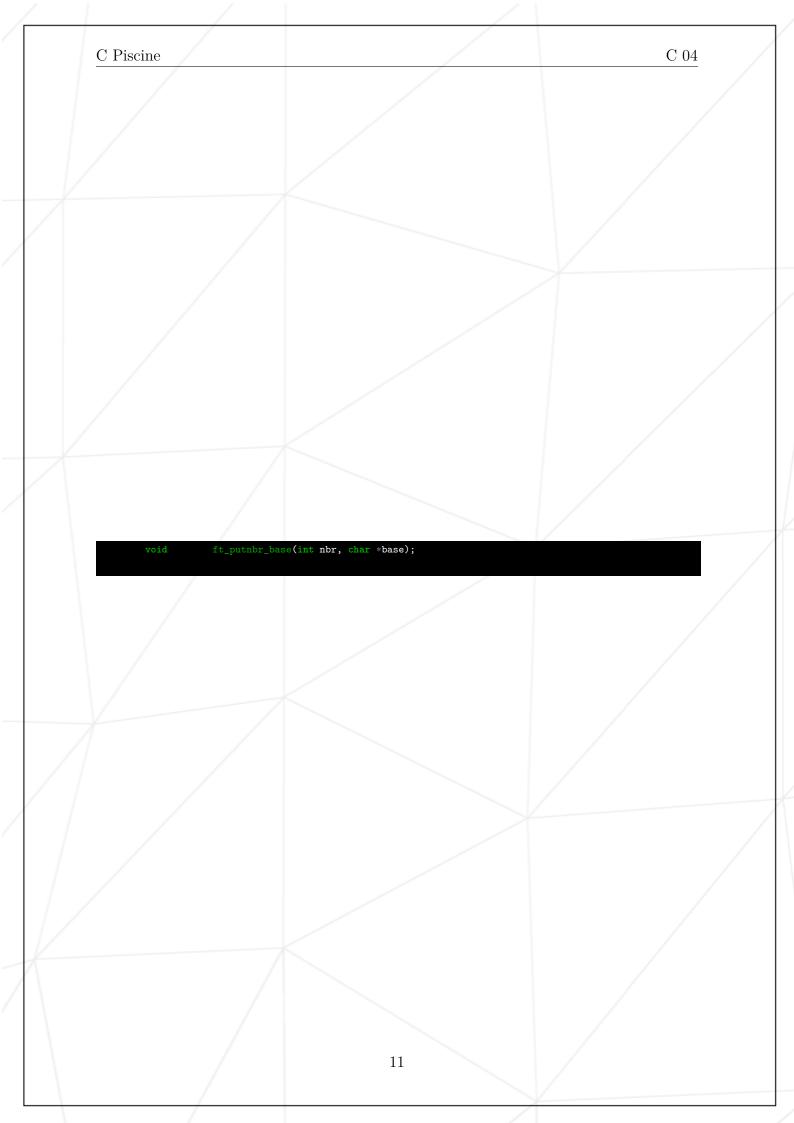
```
int ft_atoi(char *str);
```

### Chapter VII

## Exercise 04: ft\_putnbr\_base

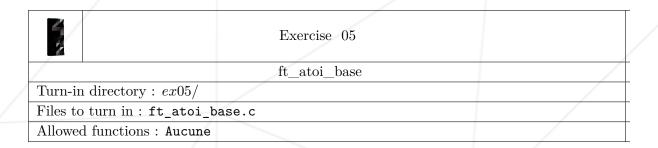
	Exercise 04	
	ft_putnbr_base	
Turn-in directory : $ex04/$		
Files to turn in : ft_putnbr		
Allowed functions: write		

- Create a function that displays a number in a base system in the terminal.
- This number is given in the shape of an int, and the radix in the shape of a string of characters.
- The base-system contains all useable symbols to display that number :
  - $\circ~0123456789$  is the commonly used base system to represent decimal numbers ;
  - 01 is a binary base system;
  - 0123456789ABCDEF an hexadecimal base system;
  - o poneyvif is an octal base system.
- The function must handle negative numbers.
- If there's an invalid argument, nothing should be displayed. Examples of invalid arguments :
  - base is empty or size of 1;
  - base contains the same character twice ;
  - $\circ$  base contains + or -;
- Here's how it should be prototyped:



## Chapter VIII

Exercise 05: ft\_atoi\_base



- Write a function that converts the initial portion of the string pointed by str to int representation.
- str is in a specific base given as a second parameter.
- excepted the base rule, the function should work exactly like ft\_atoi.
- If there's an invalid argument, the function should return 0. Examples of invalid arguments:
  - base is empty or size of 1;
  - base contains the same character twice;
  - $\circ$  base contains + or or whitespaces;
- Here's how it should be prototyped:

int ft\_atoi\_base(char \*str, char \*base);