



C Piscine

C 01

*Summary:* このドキュメントは、*C Piscine @ 42*の *C 01*モジュール用の課題である。

# 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_ft</b>	<b>5</b>
<b>IV</b>	<b>Exercise 01 : ft_ultimate_ft</b>	<b>6</b>
<b>V</b>	<b>Exercise 02 : ft_swap</b>	<b>7</b>
<b>VI</b>	<b>Exercise 03 : ft_div_mod</b>	<b>8</b>
<b>VII</b>	<b>Exercise 04 : ft_ultimate_div_mod</b>	<b>9</b>
<b>VIII</b>	<b>Exercise 05 : ft_putstr</b>	<b>10</b>
<b>IX</b>	<b>Exercise 06 : ft_strlen</b>	<b>11</b>
<b>X</b>	<b>Exercise 07 : ft_rev_int_tab</b>	<b>12</b>
<b>XI</b>	<b>Exercise 08 : ft_sort_int_tab</b>	<b>13</b>
<b>XII</b>	<b>Submission and peer-evaluation</b>	<b>14</b>

# Chapter I

## Instructions

- 課題に関する噂に惑わされないよう気をつけ、信用しないこと。
- この書類は、提出前に変更になる可能性があるため、気をつけること。
- ファイルとディレクトリへの権限があることを、あらかじめ確認すること。
- すべての課題は、提出手順に従い行うこと。
- 課題の確認と評価は、あなたの周りにいるPiscine受験者により行われる。
- 課題の確認と評価は、Piscine受験者に加えて、Moulinetteと呼ばれるプログラムによっても行われる。
- Moulinetteは、大変細かい評価を行う。これはすべて自動で行われるため、交渉の余地はない。
- Moulinetteは、コーディング規範（Norm）を遵守しないコードを解釈することができない。そのため、Moulinetteはnorminetteと呼ばれるプログラムを使用し、あなたのファイルがコーディング規範を遵守しているか確認を行う。せっかくの取り組みが、norminetteの確認により無駄にならないよう、気をつけること。
- 問題は、簡単なものから徐々に難しくなるように並べられている。簡単な問題が解けていない場合は、難しい問題が解けていたとしても 加点されることはない。
- 使用が禁止されている関数を使用した場合は、不正とみなされる。不正者は-42の評価をつけられ、この評価に対する交渉の余地はない。
- 課題がプログラムの提出を要求する場合のみ、main()関数を提出すること。
- Moulinetteは以下のフラッグを用いて、ccでコンパイルする。 -Wall -Wextra -Werror
- プログラムがコンパイルされなかった場合、評価は0になる。
- 課題で指定されていないものは、どんなファイルもディレクトリ内に置かないこと。

- 質問がある場合は、隣の人に聞くこと。それでも分からない場合は、反対側の席の人に聞くこと。
- 助けてくれるのは、Google / 人間 / インターネット / ...と呼ばれているものたちである。
- 出力例には、問題文に明記されていない細部まで表示されている場合があるため、入念に確認すること。



norminetteは、`-R CheckForbiddenSourceHeader` をオプションに追加しなければならない。Moulinetteも、このオプションを使用する。

# Chapter II

## 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 you with a Royale with cheese.

# Chapter III

## Exercise 00 : ft\_ft


	Exercise 00
ft_ft	
提出するディレクトリ : <i>ex00/</i>	
提出するファイル : <i>ft_ft.c</i>	
使用可能な関数 : None	

- パラメータとして、intへのポインタを受け取り、そのintの値を42にする関数を作成せよ。
- プロトタイプ例)

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

# Chapter IV

## Exercise 01 : ft\_ultimate\_ft


	Exercise 01
	ft_ultimate_ft
	提出するディレクトリ : <i>ex01/</i>
	提出するファイル : <i>ft_ultimate_ft.c</i>
	使用可能な関数 : None

- パラメータとして、intへのポインタのポインタのポインタのポインタのポインタのポインタのポインタのポインタのポインタのポインタを受け取り、そのintの値を42にする関数を作成せよ。
- プロトタイプ例)

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

# Chapter V

## Exercise 02 : ft\_swap

	Exercise 02
	ft_swap
提出するディレクトリ : <i>ex02/</i>	
提出するファイル : <b>ft_swap.c</b>	
使用可能な関数 : None	


- パラメータとして渡された2つの整数の値を交換する関数を作成せよ。
- プロトタイプ例)

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



# Chapter VI

## Exercise 03 : ft\_div\_mod

	Exercise 03
	ft_div_mod
	提出するディレクトリ : <i>ex03/</i>
	提出するファイル : <i>ft_div_mod.c</i>
	使用可能な関数 : None


- ft\_div\_mod 関数を作成せよ。

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

- この関数は、パラメータaをbで割り、その商をdivに指定されたintに格納する。  
割り算の余りは、modに指定されたintに格納する。

# Chapter VII

## Exercise 04 : ft\_ultimate\_div\_mod

	Exercise 04
ft_ultimate_div_mod	
提出するディレクトリ : ex04/	
提出するファイル : ft_ultimate_div_mod.c	
使用可能な関数 : None	


- 以下のようなプロトタイプの関数 ft\_ultimate\_div\_mod を作成せよ。

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

- この関数は、パラメーターaをbで割り、その商をaに指定されたintに格納する。  
割り算の余りは、bに指定されたintに格納する。

# Chapter VIII

## Exercise 05 : ft\_putstr


	Exercise 05
	ft_putstr
	提出するディレクトリ : <i>ex05/</i>
	提出するファイル : <i>ft_putstr.c</i>
	使用可能な関数 : <i>write</i>

- 文字列を標準出力に出力する関数を作成せよ。
- プロトタイプ例)

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

# Chapter IX

## Exercise 06 : ft\_strlen


	Exercise 06
	ft_strlen
	提出するディレクトリ : <i>ex06/</i>
	提出するファイル : <i>ft_strlen.c</i>
	使用可能な関数 : None

- 文字列の文字の数を返す関数を作成せよ。
- プロトタイプ例)

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

# Chapter X

## Exercise 07 : ft\_rev\_int\_tab


	Exercise 07
	ft_rev_int_tab
	提出するディレクトリ : <i>ex07/</i>
	提出するファイル : <i>ft_rev_int_tab.c</i>
	使用可能な関数 : None

- 与えられた整数の配列の順を逆にして返す関数を作成せよ。
- 引数は、intへのポインタと配列の要素の数である。
- プロトタイプ例)

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

# Chapter XI

## Exercise 08 : ft\_sort\_int\_tab

	Exercise 08
	ft_sort_int_tab
	提出するディレクトリ : <i>ex08/</i>
	提出するファイル : <i>ft_sort_int_tab.c</i>
	使用可能な関数 : None

- 整数の配列を、昇順に並び替える関数を作成せよ。
- 引数は、intへのポインタと配列の要素の数である。
- プロトタイプ例)

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

# Chapter XII

## Submission and peer-evaluation

課題は、いつも通り Git リポジトリに提出すること。リポジトリ内の提出物のみが、レビュー中の評価対象となる。ファイルの名前が正しいことを確認すること。



この課題の要件で求められているファイルのみを提出すること。