



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す。

# 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>

# Chapter I

## Instructions

- このページのみを参考にしてください。噂を信用しないで下さい。
- この書類は、提出前に変更になる可能性があります。十分に注意して下さい。
- 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して下さい。
- 課題は全て提出手順に従って行って下さい。
- 課題の確認と評価は、あなたのクラスメイトが行います。
- 課題はMoulinetteと呼ばれるプログラムによっても確認・評価されます。
- Moulinetteは大変細かい評価を行います。全て自動で行われ、交渉方法はありません。頑張ってください。
- Moulinetteは規範を無視したコードは解読できません。Moulinetteはあなたのファイルが規範を遵守しているかをチェックするために、norminetteと呼ばれるプログラムを使って判断します。要約：せっかくの取り組みがnorminetteのチェックによって無駄になるのは勿体無いので、気をつけましょう。
- 課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ます。簡単な課題が解けていない場合、難しい問題かが解けていたとしても **加点されることはありません**。
- 禁止されている関数をしようした場合は不正とみなします。不正者は-42の評価をつけられこの評価に交渉の余地はありません。
- プログラムを要求する際はmain()関数のみを提出しましょう。
- Moulinetteはこれらのフラグを用いてgccでコンパイルします：-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です。
- 課題で指定されているもの以外はどんなファイルもディレクトリ内に残しておくことはできません。
- 質問があれば右側の人に聞きましょう。それでも分からなければ左側の人に聞いてください。

- あなたを助けてくれるのはGoogle / 人間 / インターネット / ...と呼ばれているものです。
- intranet上のフォーラムの” C Piscine” パートかPiscineのslackを確認してください。
- 例を徹底的に調べてください。課題で言及されていない詳細まで要求されます。
- By Odin, by Thor! 頭を使って!!!



Norminetteは、 `-R CheckForbiddenSourceHeader` をオプションに追加しなければなりません。その際、Moulinetteも使用します。

# 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 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にする関数を作成しましょう。
- 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を受け取り、そのintの値を42にする関数を作成しましょう。
- 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 : <i>ft_swap.c</i>
	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りましょう。
- 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 関数を作成しましょう。

```
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で割ります。割り算の結果はdivに指定されたintに格納されます。 aをbで割られた余りはmodに指定されたintに格納されます。

# 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


- 下記のようなプロトタイプの関数 ft\_ultimate\_div\_mod を作成しましょう。

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

- この関数はパラメーターaをbで割ります。割り算の結果はaに指定されたintに格納されます。割り算の余りはbに指定されたintに格納されます。

# 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>

- 文字列を標準出力する関数を作成しましょう。
- 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

- 文字列内の文字数を数えて返す関数を作成しましょう。
- 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しましょう。
- 引数はintへのポインターと配列内の要素の数です。
- 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

- 整数の配列を昇順で並べかえる関数を作成しましょう。
- 引数はintへのポインタと配列内の要素の数です。
- プロトタイプ例

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