



C Piscine

C 00

*Summary:* このドキュメントは、*C Piscine @ 42*の *C 00*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_putchar</b>	<b>5</b>
<b>IV</b>	<b>Exercise 01 : ft_print_alphabet</b>	<b>6</b>
<b>V</b>	<b>Exercise 02 : ft_print_reverse_alphabet</b>	<b>7</b>
<b>VI</b>	<b>Exercise 03 : ft_print_numbers</b>	<b>8</b>
<b>VII</b>	<b>Exercise 04 : ft_is_negative</b>	<b>9</b>
<b>VIII</b>	<b>Exercise 05 : ft_print_comb</b>	<b>10</b>
<b>IX</b>	<b>Exercise 06 : ft_print_comb2</b>	<b>11</b>
<b>X</b>	<b>Exercise 07 : ft_putnbr</b>	<b>12</b>
<b>XI</b>	<b>Exercise 08 : ft_print_combn</b>	<b>13</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は以下のフラグを用いて、gcc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

タラ肝油は、タラの肝臓に由来する栄養補助食品です。

ほとんどの魚油と同様に、オメガ-3脂肪酸、  
エイコサペンタエン酸（EPA）、ドコサヘキサエン酸（DHA）が豊富に含まれています。

タラ肝油には、ビタミンAとビタミンDも含まれています。


歴史的には、ビタミンAとビタミンDの補給のために摂取されてきました。

ビタミンDは、くる病やその他のビタミンD欠乏症の症状を予防することが示されているため、  
かつては子供に一般的に投与されていました。

タラ肝油とは違い、Cはとても良い味です。積極的に摂取しましょう！

# Chapter III

## Exercise 00 : ft\_putchar

	Exercise 00
	ft_putchar
	提出するディレクトリ : <i>ex00/</i>
	提出するファイル : <i>ft_putchar.c</i>
	使用可能な関数 : <i>write</i>

- パラメータとして渡された文字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_putchar(char c);
```

文字を標準出力に出力するためには、以下のようにwrite関数を使うこと。

```
write(1, &c, 1);
```

# Chapter IV

## Exercise 01 : ft\_print\_alphabet


	Exercise 01
	ft_print_alphabet
	提出するディレクトリ : <i>ex01/</i>
	提出するファイル : <i>ft_print_alphabet.c</i>
	使用可能な関数 : <i>write</i>

- 小文字のアルファベットすべてを、“a” から昇順に1行で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_print_alphabet(void);
```

# Chapter V

## Exercise 02 : ft\_print\_reverse\_alphabet

	Exercise 02
	ft_print_reverse_alphabet
	提出するディレクトリ : <i>ex02/</i>
	提出するファイル : <i>ft_print_reverse_alphabet.c</i>
	使用可能な関数 : <i>write</i>


- 小文字のアルファベットすべてを、“z” から降順に1行で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_print_reverse_alphabet(void);
```



# Chapter VI

## Exercise 03 : ft\_print\_numbers


	Exercise 03
	ft_print_numbers
	提出するディレクトリ : <i>ex03/</i>
	提出するファイル : <i>ft_print_numbers.c</i>
	使用可能な関数 : <i>write</i>

- すべての数字を、昇順に1行で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_print_numbers(void);
```

# Chapter VII

## Exercise 04 : ft\_is\_negative


	Exercise 04
	ft_is_negative
提出するディレクトリ : <i>ex04/</i>	
提出するファイル : <i>ft_is_negative.c</i>	
使用可能な関数 : <i>write</i>	

- パラメータとして入力された整数に応じて、“N” または、“P” を標準出力に出力する関数を作成せよ。 *n* がマイナスの場合は “N”、*n* がプラスまたは、*null* の場合は、“P” を標準出力に出力すること。
- プロトタイプ例)

```
void ft_is_negative(int n);
```

# Chapter VIII

## Exercise 05 : ft\_print\_comb

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

- 3桁の数字の異なるすべての組み合わせを、昇順で標準出力に出力する関数を作成せよ。
- 例)


```
$> ./a.out | cat -e
012, 013, 014, 015, 016, 017, 018, 019, 023, ..., 789$>
```

- 789が既に存在するため、987は存在しない。
- 9が複数存在するため、999は存在しない。
- プロトタイプ例)

```
void ft_print_comb(void);
```

# Chapter IX

## Exercise 06 : ft\_print\_comb2

	Exercise 06
	ft_print_comb2
	提出するディレクトリ : <i>ex06/</i>
	提出するファイル : <i>ft_print_comb2.c</i>
	使用可能な関数 : <i>write</i>

- 00から99までの2桁の数字の異なるすべての組み合わせを、昇順で標準出力に出力する関数を作成せよ。

- 例)


```
$> ./a.out | cat -e
00 01, 00 02, 00 03, 00 04, 00 05, ..., 00 99, 01 02, ..., 97 99, 98 99$>
```

- プロトタイプ例)

```
void ft_print_comb2(void);
```

# Chapter X

## Exercise 07 : ft\_putnbr

	Exercise 07
	ft_putnbr
	提出するディレクトリ : <i>ex07/</i>
	提出するファイル : <i>ft_putnbr.c</i>
	使用可能な関数 : <i>write</i>


- パラメータとして渡された数を標準出力に出力する関数を作成せよ。その際、`int`型に収まるすべてのパターンを標準出力に出力できるようにすること。
- プロトタイプ例)

```
void ft_putnbr(int nb);
```

- 例)
  - `ft_putnbr(42)` は、“42”を標準出力に出力する。

# Chapter XI

## Exercise 08 : ft\_print\_combn

	Exercise 08
	ft_print_combn
	提出するディレクトリ : <i>ex08/</i>
	提出するファイル : <i>ft_print_combn.c</i>
	使用可能な関数 : <i>write</i>

- *n*の数の異なるすべての組み合わせを、昇順で標準出力に出力する関数を作成せよ。
- *n*は、次のようになる。 :  $0 < n < 10$
- *n* = 2の場合の例)

```
$>./a.out | cat -e
01, 02, 03, ..., 09, 12, ..., 79, 89$>
```

- プロトタイプ例)

```
void ft_print_combn(int n);
```