



C Piscine

C 08

*Summary:* このドキュメントはC Piscine @ 42の C 08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.h</b>	<b>5</b>
<b>IV</b>	<b>Exercise 01 : ft_boolean.h</b>	<b>6</b>
<b>V</b>	<b>Exercise 02 : ft_abs.h</b>	<b>8</b>
<b>VI</b>	<b>Exercise 03 : ft_point.h</b>	<b>9</b>
<b>VII</b>	<b>Exercise 04 : ft_strs_to_tab</b>	<b>10</b>
<b>VIII</b>	<b>Exercise 05 : ft_show_tab</b>	<b>12</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はこれらのフラッグを用いて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。

# Chapter II

## Foreword

Here's what Wikipedia have to say about Platypus :

The platypus (*Ornithorhynchus anatinus*), also known as the duck-billed platypus, is a semiaquatic egg-laying mammal endemic to eastern Australia, including Tasmania. Together with the four species of echidna, it is one of the five extant species of monotremes, the only mammals that lay eggs instead of giving birth. The animal is the sole living representative of its family (*Ornithorhynchidae*) and genus (*Ornithorhynchus*), though a number of related species have been found in the fossil record.


The unusual appearance of this egg-laying, duck-billed, beaver-tailed, otter-footed mammal baffled European naturalists when they first encountered it, with some considering it an elaborate hoax. It is one of the few venomous mammals, the male platypus having a spur on the hind foot that delivers a venom capable of causing severe pain to humans. The unique features of the platypus make it an important subject in the study of evolutionary biology and a recognisable and iconic symbol of Australia; it has appeared as a mascot at national events and is featured on the reverse of its 20-cent coin. The platypus is the animal emblem of the state of New South Wales.

Until the early 20th century, it was hunted for its fur, but it is now protected throughout its range. Although captive breeding programs have had only limited success and the platypus is vulnerable to the effects of pollution, it is not under any immediate threat.

This subject is absolutly not talking about platypus.

# Chapter III

## Exercise 00 : ft.h


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

- *ft.h* ファイルを作成しましょう。
- これらの関数のプロトタイプをファイルにいらしてください。

```
void    ft_putchar(char c);
void    ft_swap(int *a, int *b);
void    ft_putstr(char *str);
int     ft_strlen(char *str);
int     ft_strcmp(char *s1, char *s2);
```

# Chapter IV

## Exercise 01 : ft\_boolean.h

	Exercise 01
	ft_boolean.h
	提出するディレクトリ : ex01/
	提出するファイル : ft_boolean.h
	使用可能な関数 : None

- ft\_boolean.h ファイルを作成しましょう。以下のファイルが適切にコンパイルされ、求めているプログラムのアウトプットを参考に作成しましょう。

```
#include "ft_boolean.h"

void ft_putstr(char *str)
{
    while (*str)
        write(1, str++, 1);
}

t_bool ft_is_even(int nbr)
{
    return ((EVEN(nbr)) ? TRUE : FALSE);
}

int main(int argc, char **argv)
{
    (void)argv;
    if (ft_is_even(argc - 1) == TRUE)
        ft_putstr(EVEN_MSG);
    else
        ft_putstr(ODD_MSG);
    return (SUCCESS);
}
```

- プログラム表示例

```
I have an even number of arguments.
```

- それか

```
I have an odd number of arguments.
```

- 適宜、改行を行なってください。




`-R CheckDefine` フラグを使用して `Norminette` を実行してください。  
`Moulinette` も同じ挙動をします。



# Chapter V

## Exercise 02 : ft\_abs.h

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

- 引数を絶対値に変換するマクロ、ABS を作成しましょう。


```
#define ABS(Value)
```



-R *CheckDefine* フラグを使用して *Norminette* を実行してください。  
*Moulinette* も同じ挙動をします。

# Chapter VI

## Exercise 03 : ft\_point.h

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

- 下記のファイルをコンパイルするために必要なft\_point.hを作成しましょう。

```
#include "ft_point.h"


void      set_point(t_point *point)
{
    point->x = 42;
    point->y = 21;
}

int       main(void)
{
    t_point      point;

    set_point(&point);
    return (0);
}
```

# Chapter VII

## Exercise 04 : ft\_strs\_to\_tab

	Exercise 04
	ft_strs_to_tab
	提出するディレクトリ : ex04/
	提出するファイル : ft_strs_to_tab.c
	使用可能な関数 : malloc, free

- 文字列の配列と、その配列のサイズを引数として受け取る関数を作成しましょう。
- プロトタイプ例

```
struct s_stock_str *ft_strs_to_tab(int ac, char **av);
```

- avの各要素を構造へと変換します。
- 構造はft\_stock\_str.hファイル内で定義されています。


```
typedef struct s_stock_str
{
    int size;
    char *str;
    char *copy;
} t_stock_str;
```

- sizeは文字列の長さです。
- strは文字列です。
- copyは文字列のコピーです。
- avの順番を維持しましょう。

- 返ってきた配列は動的に確保されたメモリに配置され、最後の要素のstrは0に設定されます。これが列の最後を意味しています。
- エラーが発生した場合は NULLポインタを返さない。
- この関数をft\_show\_tab(次の問題)でテストしますので、それに応じて動くようにしてください!

# Chapter VIII

## Exercise 05 : ft\_show\_tab

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

- 前回の関数によって作られた配列の要素を表示する関数を作成しましょう。
- プロトタイプ例

```
void ft_show_tab(struct s_stock_str *par);
```

- 構造は前回の問題と同様で、ft\_stock\_str.h ファイル内で定義づけられるでしょう。
- 各要素の表示例
  - 文字列と'\n'
  - 文字列のサイズと'\n'
  - 文字列のコピー（修正の可能性あり）と'\n'
- この関数を私たちのft\_strs\_to\_tab (前の課題)でテストしますので、それに応じて動くようにしてください!