



C Piscine

C 08

Summary: このドキュメントは、42の *C Piscine C 08* モジュール用の課題です。

Version: 14

Contents

I	Instructions	2
II	AI Instructions	4
III	Foreword	7
IV	Exercise 00 : ft.h	8
V	Exercise 01 : ft_boolean.h	9
VI	Exercise 02 : ft_abs.h	11
VII	Exercise 03 : ft_point.h	12
VIII	Exercise 04 : ft_strs_to_tab	13
IX	Exercise 05 : ft_show_tab	15
X	Submission and peer-evaluation	16

Chapter I

Instructions

- このページのみが正式な課題の指示です。噂に惑わされないようにしてください。
- この課題は提出前に変更される可能性がありますので、気をつけてください！
- ファイルとディレクトリへの適切な権限があることを確認してください。
- すべての課題で、提出手順に従ってください。
- 提出した課題の評価（レビュー）は、あなたの周りにはいるPiscine受験者によって行われます。
- さらに、**Moulinette**（自動採点プログラム）にも評価されます。
- **Moulinette**は非常に厳格かつ厳密に評価を行います。完全に自動化されており、交渉の余地はありません。思わぬ減点を避けるためにも、細部まで丁寧に取り組んでください。
- **Moulinette**は融通が利きません。あなたのコードがNormに準拠していない場合、理解しようとしません。**Moulinette**は**norminette**というプログラムを使用して、ファイルがNormに準拠しているかチェックします。要約：**norminette**のチェックに通らない課題を提出しても意味がありません。
- 課題は難易度が低いものから高いものへと難易度順に並んでいます。前の課題が正しく解けていない場合、後の課題が解けていても評価されません。
- 禁止されている関数を使用することは不正行為とみなされます。不正行為者は**-42**という成績がつけられ、これは交渉の余地がありません。
- 私たちが具体的にプログラムを求める場合のみ、**main()**関数を提出する必要があります。
- **Moulinette**は以下のフラグでコンパイルします：**-Wall -Wextra -Werror, cc**を使用します。
- プログラムがコンパイルできない場合、**0**の評価になります。
- 課題で指定されたファイル以外は、ディレクトリに絶対に残さないでください。

- 質問がある場合は、まず右隣の仲間に聞いてみましょう。いなければ左隣の仲間に聞いてみてください。
- あなたの助けになる参考資料はGoogle / man / インターネット / ...です。
- Intranetのフォーラムの「C Piscine」セクションをチェックしてください。
- 課題の例（出力例）をよく読んでください。課題の説明文だけではすぐに分からない要件が隠れていることがあります。
- 頭を使ってください!!!

Chapter II

AI Instructions

● Context

C Piscineは強烈な体験です。42で迎える最初の大きな挑戦として、問題解決、自律性、そしてコミュニティの世界に深く飛び込むことになります。

このフェーズの主な目的は、試行錯誤や反復、そして特にピアラーニングを通じた交流によって、自分の土台を築き上げることです。

AI時代において、近道を見つけることは簡単です。しかし、AIの利用が本当にあなたの成長を助けているのか、それとも真のスキルを身につける上で障害となっていないか、よく考えることが大切です。

Piscineはまたとない人との繋がりを経験する場でもあります。今のところ、それにとって代わるものは何ともありません。AIでさえも。

学習ツールとして、ICTカリキュラムの一環として、そして労働市場での期待の高まりに応えるものとして、私たちのAIに対する考え方の詳細については、Intranet上の専用FAQをご覧ください。

● 主なメッセージ

- 近道をせず、強固な基礎を築く。
- 技術力と実践力を真に養う。
- 真のピアラーニングを体験し、学び方や新たな問題の解決方法を学び始める。
- 結果よりも学習の過程が重要である。
- AIに伴うリスクを学び、一般的な落とし穴を避けるための効果的な管理方法と対策を身につける。

● 学習者のルール:

- 課題には、AIに頼る前にまず自分の頭で考えること。
- AIに直接的な答えを求めないこと。
- AIに対する42のグローバルな方針を学ぶこと。

● このフェーズでの到達目標:

この基礎フェーズを終えることで、以下の点を達成できます。

- 技術とコーディングの適切な基礎を身につける。
- この段階においてAIが危険となりうる理由と、その危険性を理解する。

● コメントと具体例:

- 私たちはAIの存在を知っていますし、それがあなたのプロジェクトを解決できることも知っています。しかし、あなたは学ぶためにここにいるのであって、AIが学習したことを証明するためではありません。AIが特定の問題を解決できることを示すためだけに、あなた（や私たち）の時間を無駄にしないでください。
- 42での学習は、答えを知ることではありません。答えを見つけ出す能力を養うことです。AIは直接答えを与えてくれますが、それはあなた自身の論理的思考を妨げます。そして、論理的思考には時間と努力、そして失敗が伴います。成功への道は、決して楽なものではありません。
- Exam中はAIが利用できないことを心に留めておいてください。インターネットもスマートフォンもありません。学習過程でAIに過度に頼りすぎている場合、すぐにそのことに気づくでしょう。
- ピアラーニングは、多様な考え方やアプローチに触れる機会を与え、対人スキルや多角的な思考能力を向上させます。これは、単にボットとチャットするよりもはるかに価値があります。ですから、恥ずかしがらずに、話しかけ、質問し、共に学びましょう！
- AIは学習ツールとして、またそれ自体がトピックとして、カリキュラムの一部になります。独自のAIソフトウェアを開発する機会さえあります。私たちの段階的なアプローチについてさらに学ぶためには、Intranetで利用可能なドキュメントを参照してください。

✓ 良い実践例:

新しいコンセプトの学習でつまづいています。近くの人にどうアプローチしたか尋ねてみます。10分ほど話していると、突然ひらめき、理解できました。

✕ 悪い実践例:

こっそりAIを使い、良さそうなコードをコピーします。ピアレビューの際には何も説明できず、不合格になりました。ExamではAIがなく、また行き詰まり、不合格になりました。

Chapter III

Foreword

Here's what Wikipedia has 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 absolutely not talking about platypuses.

Chapter IV

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 : ft.h
	使用可能な関数 : None

- ft.hファイルを作成してください。
- 以下のすべての関数のプロトタイプをファイルに入れてください。

```
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 V

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ファイルを作成してください。このファイルは、以下のmain関数が正しくコンパイル・実行できるようにしてください。

```
#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);
}
```

- プログラムの出力例 :

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

- 上記、もしくは、以下のように標準出力に出力されます。

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


- 必要に応じて改行を末尾に追加してください。



Norminetteは、`-R CheckDefine` をオプションに追加しなければなりません。
Moulinetteも同様にこのオプションを使用します。

Chapter VI

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 : <i>ft_abs.h</i>
	使用可能な関数 : <i>None</i>

- 引数を絶対値に変換するマクロABSを作成してください。


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



Norminetteは、`-R CheckDefine` をオプションに追加しなければなりません。
Moulinetteも同様にこのオプションを使用します。

Chapter VII

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

- 以下のmain関数をコンパイルするために必要なft_point.hファイルを作成してください。

```
#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 VIII

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、そのサイズを引数として受け取る関数を作成してください。
- プロトタイプ例 :

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

- この関数は、avの各要素を構造体に変換する必要があります。
- 構造体はft_stock_str.hファイルの中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の順序を保持してください。
- 返される配列はメモリ上に動的に配置され、最後の要素のstrは0にしてください。(配列の終端を示します。)

- エラーが発生した場合は、NULLを返してください。
- この関数は、次の問題のft_show_tab関数を使用してテストされます。その仕様に従って動作するように実装してください。

Chapter IX

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>

- 前問の関数で作成した配列の要素を、標準出力に出力する関数を作成してください。
- プロトタイプ例 :

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

- 構造体は前の課題と同じく、*ft_stock_str.h*ファイルの中で定義されています。
- 各要素の出力例 :
 - 文字列と “\n”
 - 文字列のサイズと “\n”
 - 文字列のコピー（修正されている可能性があるもの）と “\n”
- この関数は、前問の*ft_strs_to_tab*関数を使用してテストされます。その仕様に従って動作するように実装してください。

Chapter X

Submission and peer-evaluation

課題は、いつも通りGitリポジトリに提出してください。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ります。ファイル名が正しいかどうか、必ず確認してください。



この課題の指示で、明示的に求められたファイルのみを提出してください。