



C Piscine

C 09

*Summary:* このドキュメントは、*C Piscine @ 42*の *C 09*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 : libft</b>    | <b>5</b> |
| <b>IV</b>  | <b>Exercise 01 : Makefile</b> | <b>6</b> |
| <b>V</b>   | <b>Exercise 02 : ft_split</b> | <b>8</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読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

Dialog from the movie The Big Lebowski:

The Dude: Walter, ya know, it's Smokey, so his toe slipped over the line a little, big deal. It's just a game, man.

Walter Sobchak: Dude, this is a league game, this determines who enters the next round robin. Am I wrong? Am I wrong?

Smokey: Yeah, but I wasn't over. Gimme the marker Dude, I'm marking it 8.

Walter Sobchak: [pulls out a gun] Smokey, my friend, you are entering a world of pain.

The Dude: Walter...

Walter Sobchak: You mark that frame an 8, and you're entering a world of pain.

Smokey: I'm not...

Walter Sobchak: A world of pain.

Smokey: Dude, he's your partner...

Walter Sobchak: [shouting] Has the whole world gone crazy? Am I the only one around here who gives a shit about the rules? Mark it zero!

The Dude: They're calling the cops, put the piece away.

Walter Sobchak: Mark it zero!

[points gun in Smokey's face]

The Dude: Walter...


Walter Sobchak: [shouting] You think I'm fucking around here? Mark it zero!

Smokey: All right, it's fucking zero. Are you happy, you crazy fuck?

Walter Sobchak: ...It's a league game, Smokey.

# Chapter III

## Exercise 00 : libft

|  |             |
|--|-------------|
|   | Exercise 00 |
| libft  |             |
| 提出するディレクトリ : <i>ex00/</i>  |             |
| 提出するファイル : <i>libft_creator.sh</i> , <i>ft_putchar.c</i> , <i>ft_swap.c</i> , <i>ft_putstr.c</i> , <i>ft_strlen.c</i> , <i>ft_strcmp.c</i> |             |
| 使用可能な関数 : <i>write</i>   |             |

- *libft.a*と呼ばれるftライブラリを作成せよ。
- *libft\_creator.sh*と呼ばれるシェルスクリプトは、ソースファイルを適切にコンパイルし、ライブラリを作成する。
- このライブラリには、以下の関数がすべて格納されていること。


```
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);
```

- 以下のコマンドラインを実行すること。

```
sh libft_creator.sh
```

# Chapter IV

## Exercise 01 : Makefile

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

- `libft.a`ライブラリをコンパイルする、`Makefile`を作成せよ。
- `Makefile`は、実行しているすべてのコマンドを標準出力に出力すること。
- `Makefile`は、不要なコマンドを一切実行しないこと。
- `Makefile`は、“`srcs`”ディレクトリからソースファイルを取得する。
- `ft_putchar.c`、`ft_swap.c`、`ft_putstr.c`、`ft_strlen.c`、`ft_strcmp.c`のソースファイルを、“`srcs`”ディレクトリに入れること。
- `Makefile`は、“`includes`”ディレクトリからヘッダーファイルを取得する。
- `ft.h`を"`includes`"ディレクトリに入れること。
- `-Wall -Wextra -Werror`フラグをこの順序で使用し、上記の`.c`を`gcc`にコンパイルすること。
- `lib`は、この問題のディレクトリのルートに置くこと。
- `.o`は`.c`の近くに置くこと。
- `Makefile`に、次のルールを実装すること。: `clean`, `fclean`, `re`, `all`, `libft.a`
- `make`は、`make all`の実行内容と同様である。
- `all`のルールは、`make libft.a`と同様である。
- `clean`のルールは、一時的に生成されたファイルをすべて削除する。

- `fclean`のルールは、`make clean`と`make all` で作成されたすべてのバイナリを削除する。
- `re`のルールは、`make fclean`のルールの後に、`make all`のルールを実行する。
- Makefileは、 ファイルをみだりにコンパイルしないこと。
- 提出されたMakefileのみがコピーされ、 テストが行われる。




ワイルドカードに注意すること。



# Chapter V

## Exercise 02 : ft\_split

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

- 文字列にある各文字に応じて、文字列を分割する関数を作成せよ。
- *charset*文字列にある各文字は、*str*文字列を分割するために使用すること。
- この関数は、*charset*文字列にある文字が*str*文字列を分割し、分割された文字列の配列を返す。
- 文字列の配列のメモリのみではなく、分割された各文字列のためにメモリを確保し、コピーすること。
- 配列内に空の文字列を格納することはできない。
- 引数として与えられた文字列は、編集することはできない。
- プロトタイプ例)

```
char **ft_split(char *str, char *charset);
```