



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す。

Contents

I	Instructions	2
II	Foreword	4
III	Exercise 00 : libft	5
IV	Exercise 01 : Makefile	6
V	Exercise 02 : ft_split	8

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。



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

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*という名前のftライブラリを作成しましょう。
- *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、ライブラリを作成します。
- このライブラリは次の関数を全て含みます。


```
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);
```

- 次のコマンドラインを実行します。

```
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 : Makefile	
使用可能な関数 : None	

- `libft.a`ライブラリをコンパイルする**Makefile**を作成しましょう。
- `makefile`は実行しているコマンドを表示します。
- `makefile`は不要なコマンドを一切実行しません。
- **Makefile** は"`srcs`"ディレクトリからソースファイルを入手します。
- `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`"ディレクトリにいてください。
- **Makefile** は"`includes`"ディレクトリからヘッダファイルを入手します。
- `ft.h`ファイルを"`includes`"ディレクトリにいてください。
- `gcc`に`-Wall -Wextra -Werror`フラグを使用し、上記の`.c`ファイルを順番にコンパイルします。
- `lib`はこの問題のディレクトリのルートに置きましょう。
- `.o`ファイルは `.c`ファイルの近くに置きましょう。
- **Makefile** は次のルールを実装する必要もあります。 `clean`, `fclean`, `re`, `all` として `libft.a`です。
- `make`の実行は`make all`の実行と同様です。
- `all`のルールは、`make libft.a`と同様です。


- `clean`のルールは、一時的に生成されたファイルを全て削除します。
- `fclean`のルールは、`make clean`と`make all` で作成された全てのバイナリを削除します。
- `re`のルールは、`make fclean`のルールに続いて`make all`のルールを実行するのと同様です。
- `makefile`は無意味にファイルをコンパイルしてはいけません。
- 私たちは提出されたMakefileだけをコピーし、こちらのテストを行います。



ワイルドカードに注意してください!

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>

- 文字列の各文字に応じて、ある文字列を分ける関数を作成しましょう。
- *charset*文字列にある各文字は、*str*文字列を分けるために使用しましょう。
- この関数は、*charset*文字列にある文字が*str*文字列を分断し、分断された文字列の配列を返します。
- 文字列の配列のメモリ確保はもちろんのこと、分断された各文字列用にメモリを確保しコピーしてください。
- 配列内に空の文字列を置くことはできません。適宜、結論を見つけ出してください。
- 引数として与えられた文字列は修正不可です。
- プロトタイプ例

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