



C Piscine

C 11

*Summary:* このドキュメントはC Piscine @ 42の C 11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_foreach</b>	<b>6</b>
<b>IV</b>	<b>Exercise 01 : ft_map</b>	<b>7</b>
<b>V</b>	<b>Exercise 02 : ft_any</b>	<b>8</b>
<b>VI</b>	<b>Exercise 03 : ft_count_if</b>	<b>9</b>
<b>VII</b>	<b>Exercise 04 : ft_is_sort</b>	<b>10</b>
<b>VIII</b>	<b>Exercise 05 : do-op</b>	<b>11</b>
<b>IX</b>	<b>Exercise 06 : ft_sort_string_tab</b>	<b>13</b>
<b>X</b>	<b>Exercise 07 : ft_advanced_sort_string_tab</b>	<b>14</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 a little story :

(1982, California) Larry Walters of Los Angeles is one of the few to contend for the Darwin Awards and live to tell the tale. "I have fulfilled my 20-year dream," said Walters, a former truck driver for a company that makes TV commercials. "I'm staying on the ground. I've proved the thing works."

Larry's boyhood dream was to fly. But fates conspired to keep him from his dream. He joined the Air Force, but his poor eyesight disqualified him from the job of pilot. After he was discharged from the military, he sat in his backyard watching jets fly overhead.

He hatched his weather balloon scheme while sitting outside in his "extremely comfortable" Sears lawnchair. He purchased 45 weather balloons from an Army-Navy surplus store, tied them to his tethered lawnchair (dubbed the Inspiration I) and filled the four-foot diameter balloons with helium. Then, armed with some sandwiches, Miller Lite, and a pellet gun, he strapped himself into his lawnchair. He figured he would shoot to pop a few of the many balloons when it was time to descend.

Larry planned to sever the anchor and lazily float to a height of about 30 feet above the backyard, where he would enjoy a few hours of flight before coming back down. But things didn't work out quite as Larry planned.

When his friends cut the cord anchoring the lawnchair to his Jeep, he did not float lazily up to 30 feet. Instead he streaked into the LA sky as if shot from a cannon, pulled by the lift of 45 helium balloons, holding 33 cubic feet of helium each.

He didn't level off at 100 feet, nor did he level off at 1000 feet. After climbing and climbing, he leveled off at 16,000 feet.

At that height he felt he couldn't risk shooting any of the balloons, lest he unbalance the load and really find himself in trouble. So he stayed there, drifting cold and frightened with his beer and sandwiches, for more than 14 hours. He crossed the primary approach corridor of LAX, where startled Trans World Airlines and Delta Airlines pilots radioed in reports

of the strange sight.


Eventually he gathered the nerve to shoot a few balloons, and slowly descended. The hanging tethers tangled and caught in a power line, blacking out a Long Beach neighborhood for 20 minutes. Larry climbed to safety, where he was arrested by waiting members of the LAPD. As he was led away in handcuffs, a reporter dispatched to cover the daring rescue asked him why he had done it. Larry replied nonchalantly, "A man can't just sit around."

The Federal Aviation Administration was not amused. Safety Inspector Neal Savoy said, "We know he broke some part of the Federal Aviation Act, and as soon as we decide which part it is, a charge will be filed."

The moral of this story is Larry Walters should have stay on his chair and learn C....

# Chapter III

## Exercise 00 : ft\_foreach

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

- 与えられた配列の各int型の整数をf関数ポインタに与えるft\_foreachを作成しましょう。
- この関数は配列順に適用されます。
- プロトタイプ例


```
void      ft_foreach(int *tab, int length, void(*f)(int));
```

- 例えば、下記のようにft\_foreach関数を呼ぶとint型の配列の全ての要素を表示します。

```
ft_foreach(tab, 1337, &ft_putnbr);
```

# Chapter IV

## Exercise 01 : ft\_map

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


- 与えられた配列の各int型の整数をf関数ポインタに適用し、全ての戻り値を保持している配列を返す、ft\_map関数を作成しましょう。
- この関数は配列順に適用されます。
- プロトタイプ例

```
int *ft_map(int *tab, int length, int(*f)(int));
```



# Chapter V

## Exercise 02 : ft\_any

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


- 配列の各要素をf関数ポインタに与えて、少なくとも1つの要素の戻り値が0以外を返す場合は1を返し、そうでなければ0を返す、ft\_any関数を作成しましょう。
- この関数は配列順に適用されます。
- プロトタイプ例

```
int ft_any(char **tab, int(*f)(char*));
```

- 配列の最後にはヌルポインタがあります。

# Chapter VI

## Exercise 03 : ft\_count\_if


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

- 配列の各要素をf関数ポインタに与えて、0以外を返した回数を返す、ft\_count\_if関数を作成しましょう。
- この関数は配列順に適用されます。
- プロトタイプ例

```
int ft_count_if(char **tab, int length, int(*f)(char*));
```

# Chapter VII

## Exercise 04 : ft\_is\_sort


	Exercise 04
	ft_is_sort
	提出するディレクトリ : <i>ex04/</i>
	提出するファイル : <i>ft_is_sort.c</i>
	使用可能な関数 : None

- 配列が昇順の場合は1を、そうでない場合には0を返す、ft\_is\_sort関数を作成しましょう。
- 引数として与えられている関数は、最初の引数が二番目の引数より小さい場合は負の整数を返し、二番目の引数と同じ場合は0、それ以外の場合は正の整数を返します。
- プロトタイプ例

```
int ft_is_sort(int *tab, int length, int(*f)(int, int));
```

# Chapter VIII

## Exercise 05 : do-op

	Exercise 05
	do-op
	提出するディレクトリ : <i>ex05/</i>
	提出するファイル : プログラムを作成するのに必要なファイル
	使用可能な関数 : <b>write</b>

- do-opと呼ばれるプログラムを作成しましょう。
- このプログラムは3つのコマンドライン引数を受け取って実行します。: do-op value1 operator value2
- 例

```
$>./do-op 42 "+" 21
63
$>
```

- operator(演算子)に比例する関数を関数ポインタの配列から選び使用しましょう。
- 無効の演算子の場合、プログラムは0を表示します。
- コマンドライン引数の数が無効の場合、do-opは何も表示しません。
- プログラムは '+' '-' '/' '\*' と '%' の記号を演算子として取り扱い結果を表示します。
- 0で割られる場合、次のように表示されます。

```
Stop : division by zero
```

- 0で除算する場合、次のように表示されます。


```
Stop : modulo by zero
```

- 参考にMoulinetteのテスト

```
$> make clean
$> make
$> ./do-op
$> ./do-op 1 + 1
2
$> ./do-op 42amis - ---20toto12
62
$> ./do-op 1 p 1
0
$> ./do-op 1 + toto3
1
$>
$> ./do-op toto3 + 4
4
$> ./do-op foo plus bar
0
$> ./do-op 25 / 0
Stop : division by zero
$> ./do-op 25 % 0
Stop : modulo by zero
$>
```

# Chapter IX

## Exercise 06 : ft\_sort\_string\_tab


	Exercise 06
	ft_sort_string_tab
	提出するディレクトリ : <i>ex06/</i>
	提出するファイル : <i>ft_sort_string_tab.c</i>
	使用可能な関数 : None

- tab内の文字列をascii順に表示する関数、ft\_sort\_string\_tabを作成しましょう。
- tabの最後の要素はヌルポインタです。
- 配列内のポインタを交換することにより、並び替えが実行されます。
- プロトタイプ例

```
void ft_sort_string_tab(char **tab);
```

# Chapter X

## Exercise 07 : ft\_advanced\_sort\_string\_tab

	Exercise 07
ft_advanced_sort_string_tab	
提出するディレクトリ : ex07/	
提出するファイル : ft_advanced_sort_string_tab.c	
使用可能な関数 : None	

- 配列の各文字列をcmp関数ポインタに与えて、その戻り値を元に配列を並び替える、ft\_advanced\_sort\_string\_tab関数を作成しましょう。
- 配列内のポインタを交換することにより、並び替えが実行されます。
- tabの最後の要素はヌルポインタです。
- プロトタイプ例

```
void ft_advanced_sort_string_tab(char **tab, int(*cmp)(char *, char *));
```



ft\_advanced\_sort\_string\_tab()に渡す関数ポインタをft\_strcmpにするとft\_sort\_string\_tab()と同じ結果を返します。