



C Piscine

C 04

*Summary:* このドキュメントは、C Piscine @ 42の C 04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 : ft_strlen</b>	<b>6</b>
<b>IV</b>	<b>Exercise 01 : ft_putstr</b>	<b>7</b>
<b>V</b>	<b>Exercise 02 : ft_putnbr</b>	<b>8</b>
<b>VI</b>	<b>Exercise 03 : ft_atoi</b>	<b>9</b>
<b>VII</b>	<b>Exercise 04 : ft_putnbr_base</b>	<b>10</b>
<b>VIII</b>	<b>Exercise 05 : ft_atoi_base</b>	<b>12</b>
<b>IX</b>	<b>Submission and peer-evaluation</b>	<b>13</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釈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は以下のフラッグを用いて、cc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

Here are the lyrics for City Hunter' s theme song "Moonlight Shadow":

The last time ever she saw him  
Carried away by a moonlight shadow  
He passed on worried and warning  
Carried away by a moonlight shadow.  
Lost in a riddle that Saturday night  
Far away on the other side.  
He was caught in the middle of a desperate fight  
And she couldn't find how to push through

The trees that whisper in the evening  
Carried away by a moonlight shadow  
Sing a song of sorrow and grieving  
Carried away by a moonlight shadow  
All she saw was a silhouette of a gun  
Far away on the other side.  
He was shot six times by a man on the run  
And she couldn't find how to push through

[Chorus]  
I stay, I pray  
See you in Heaven far away...  
I stay, I pray  
See you in Heaven one day.

Four A.M. in the morning  
Carried away by a moonlight shadow  
I watched your vision forming  
Carried away by a moonlight shadow  
A star was glowing in the silvery night  
Far away on the other side  
Will you come to talk to me this night  
But she couldn't find how to push through


[Chorus]

Far away on the other side.  
Caught in the middle of a hundred and five  
The night was heavy and the air was alive  
But she couldn't find how to push through  
Carried away by a moonlight shadow  
Carried away by a moonlight shadow  
Far away on the other side.

Unfortunately, this topic has nothing to do with City Hunter.

# Chapter III

## Exercise 00 : ft\_strlen


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

- 文字列にある文字の数を返す関数を作成せよ。
- プロトタイプ例)

```
int ft_strlen(char *str);
```

# Chapter IV

## Exercise 01 : ft\_putstr

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


- 標準出力に文字列を出力する関数を作成せよ。
- プロトタイプ例)

```
void    ft_putstr(char *str);
```



# Chapter V

## Exercise 02 : ft\_putnbr

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


- 引数として与えられた数字を標準出力に出力する関数を作成せよ。その際、`int`型に収まるすべてのパターンを出力すること。
- プロトタイプ例)

```
void ft_putnbr(int nb);
```

- 例)
  - `ft_putnbr(42)` displays "42".

# Chapter VI

## Exercise 03 : ft\_atoi

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

- `str`が指定する文字列の最初を、`int`型に変換する関数を作成せよ。
- 文字列の先頭に、好きな数の空白を入れることができる。( `isspace(3)` で定義される。)
- 文字列の後に、“+” や “-” を記入することができる。“-” の数が奇数もしくは、偶数であるかにより、返される整数の符号は変化する。
- 最終的に、文字列の後に10進数の数字が続いていること。
- これらのルールに従わない文字列を見つけるまで文字列を読み込み、それまでに読み込んだ整数のみを返すこと。
- オーバーフローとアンダーフローは、処理しないこと。これらの結果は、未定義となる。
- 以下に、`atoi`が返した値を標準出力に出力するプログラムの例を示す。


```
$>./a.out " ---+---+1234ab567"  
-1234
```

- プロトタイプ例)

```
int    ft_atoi(char *str);
```

# Chapter VII

## Exercise 04 : ft\_putnbr\_base


	Exercise 04
	ft_putnbr_base
	提出するディレクトリ : <i>ex04/</i>
	提出するファイル : <i>ft_putnbr_base.c</i>
	使用可能な関数 : <i>write</i>

- 整数を、ベースシステムで標準出力に出力する関数を作成せよ。
- この数字はintで与えられ、ベースは文字列で与えられる。
- ベースシステムには、数字を出力するために使用できるすべてのシンボルが含まれている。
  - 0123456789は、10進数を出力するために、一般的に使用されるベースシステムである。
  - 01 は、バイナリーベースシステムである。
  - 0123456789ABCDEF は、16進数のベースシステムである。
  - poneyvif は、8進数のベースシステムである。
- 与えられる数字は、整数であるため、負の数も含まれる。
- 無効な引数がある場合は、標準出力に何も出力しないこと。無効な引数の例)
  - ベースが空、または、1のサイズである。
  - ベースに同じ文字が2つ含まれている。
  - ベースに “+” または、“-” が含まれている。
- プロトタイプ例)

```
void      ft_putnbr_base(int nbr, char *base);
```

# Chapter VIII

## Exercise 05 : ft\_atoi\_base

	Exercise 05
	ft_atoi_base
	提出するディレクトリ : <i>ex05/</i>
	提出するファイル : <i>ft_atoi_base.c</i>
	使用可能な関数 : None

- strが指定する文字列の最初を、int型に変換する関数を作成せよ。
- strは、2つ目の引数として与えられた、特定のベースで表されている文字列である。
- ベースのルールを除いた場合、関数はft\_atoiのように働く。
- 無効な引数がある場合は、0を返すこと。無効な引数の例)
  - ベースが空、または、1のサイズである。
  - ベースに同じ文字が2つ含まれている。
  - ベースに“+”、または、“-”、もしくは空白が含まれている。
- プロトタイプ例)

```
int ft_atoi_base(char *str, char *base);
```

# Chapter IX

## Submission and peer-evaluation

課題は、いつも通り Git リポジトリに提出すること。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る。ファイルの名前が正しいことを確認すること。



この課題の要件で求められているファイルのみを提出すること。