



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_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>

# 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。



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

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

- 文字列内の文字数を数えて返す関数を作成しましょう。
- 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表示する関数を作成しましょう。
- 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>


- パラメータとして入力された数字を表示する関数を作成しましょう。その際、`int`型に収まる全てのパターンを標準出力できるようにしてください。
- 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が指す文字列の初めをint型整数に変換する関数を作成しましょう。
- 文字列は好きな量の空白で始まるのが可能です。(isspace(3)で決定)
- 文字列は任意の+や-を空白の後に綴ることが可能です。-の数が奇数か偶数によって返すべき整数の符号が変わります。
- 最後に文字列には10進数の数字が続きます。
- 文字列がこれらのルールに沿わない場合、これまでルールに沿っていた文字列だけで整数を返します。
- オーバーフローもしくはアンダーフローは処理しないようにしましょう。それらの場合、結果が未定義になっても許容します。
- atoiが返した値を表示するプログラムの例です。


```
$>./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表示する関数を作成しましょう。
- この数字はint内で与えられ、string of charactersの形は基数です。
- ベースシステムは数字を表示するために使用可能なすべてのシンボルを含んでいます。
  - 0123456789は10進数を表示するために一般的に使用されるベースシステムです。
  - 01 はバイナリーベースシステムです。
  - 0123456789ABCDEF は16進数のベースシステムです。
  - poneyvif は8進数のベースシステムです。
- 関数は負の数を扱う必要があります。
- 無効の引数があった場合は何も表示されません。無効の引数の例：
  - ベースは空か1のサイズです
  - ベースは同じ文字を2回含んでいます
  - ベースは+か-を含んでいます
- 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 : ex05/
	提出するファイル : ft_atoi_base.c
	使用可能な関数 : None

- strが指す文字列の初めの部分をint型に変換する関数を作成しましょう。
- strは2つ目のパラメータ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す。
- ベースのルールを除けば、関数はft\_atoiのように働きます。
- 無効の引数があった場合、関数は0を返します。無効の引数の例：
  - ベースは空か1のサイズです
  - ベースは同じ文字を2回含んでいます
  - ベースは+か-かスペースを含んでいます
- プロトタイプ例

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