



# C Piscine

## Shell 01

*Summary:* このドキュメントは、*C Piscine @ 42*の *Shell 01*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: Exam</b>	<b>5</b>
<b>IV</b>	<b>Exercise 01: print_groups</b>	<b>6</b>
<b>V</b>	<b>Exercise 02: find_sh</b>	<b>7</b>
<b>VI</b>	<b>Exercise 03: count_files</b>	<b>8</b>
<b>VII</b>	<b>Exercise 04: MAC</b>	<b>9</b>
<b>VIII</b>	<b>Exercise 05: Can you create it ?</b>	<b>10</b>
<b>IX</b>	<b>Exercise 06: Skip</b>	<b>12</b>
<b>X</b>	<b>Exercise 07: r_dwssap</b>	<b>13</b>
<b>XI</b>	<b>Exercise 08: add_chelou</b>	<b>14</b>
<b>XII</b>	<b>Submission and peer-evaluation</b>	<b>15</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際は、main()関数のみを提出すること。
- 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。

# Chapter II

## Foreword

Here's what Wikipedia has to say about otters :

The European otter (*Lutra lutra*), also known as the Eurasian otter, Eurasian river otter, common otter and Old World otter, is a European and Asian member of the Lutrinae or otter subfamily, and is typical of freshwater otters.

The European otter is a typical species of the otter subfamily. Brown above and cream below, these long, slender creatures are well-equipped for their aquatic habits. Its bones show osteosclerosis, increasing their density to reduce buoyancy.

This otter differs from the North American river otter by its shorter neck, broader visage, the greater space between the ears and its longer tail.

However, the European otter is the only otter in its range, so it cannot be confused for any other animal. Normally, this species is 57 to 95 cm (23-37 in) long, not counting a tail of 35-45 cm (14-18 in).

The female is shorter than the male.

The otter's average body weight is 7 to 12 kg (15.4-26.4 lbs), although occasionally a large old male may reach up to 17 kg (37 lbs).

The record-sized specimen, reported by a reliable source but not verified, weighed over 24 kg (53 lbs).

The European otter is the most widely distributed otter species, its range including parts of Asia and Africa, as well as being spread across Europe. Though currently believed to be extinct in Liechtenstein, and Switzerland, they are now very common in Latvia, along the coast of Norway and across Great Britain, especially Shetland, where 12% of the UK breeding population exist. Ireland has the highest density of Eurasian otters in Europe.


In Italy, they can be found in southern parts of the peninsula.

The South Korean population is endangered.

Otters are cute.

# Chapter III


## Exercise 00: Exam

	Exercise : 00
	Exam

- 毎週、Intranetで金曜日のExamに登録することを忘れないこと。
- Exam00というプロジェクトにも登録すること。
- Exam（イベントとプロジェクトの両方）に登録できているかを確認すること。
- Exam（イベントとプロジェクトの両方）に登録できているかを再確認すること。

# Chapter IV

## Exercise 01: print\_groups

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

- 環境変数 FT\_USER に含まれるログインが、所属しているグループのリストを標準出力に出力するコマンドラインを作成せよ。その際、スペースは入れずカンマで区切ること。
- 例)
  - FT\_USER=noursの場合、結果は “god,root,admin,master,nours,bocal”（引用符を除く）になる。

```
$>./print_groups.sh
god,root,admin,master,nours,bocal$>
```

- FT\_USER=daemonの場合、結果は “daemon,bin”（引用符を除く）になる。

```
$>./print_groups.sh
daemon,bin$>
```




man id



他のPiscine受験生に問題を解いてもらうのではなく、学び合いを通してインスピレーションを得ること。

# Chapter V

## Exercise 02: find\_sh

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

- サブディレクトリを含む現在のディレクトリ内のすべてから、名前が“.sh”（引用符を除く）で終わるファイルを探すコマンドを作成せよ。これらのファイル名を標準出力に出力する際は、.shが付記されていない状態にすること。
- 例)

```
$>./find_sh.sh | cat -e
find_sh$
file1$
file2$
file3$
$>
```




いかなる情報源も鵜呑みにはしないこと。常に自分自身でテスト・管理・検証を行うこと。



# Chapter VI

## Exercise 03: count\_files

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

- サブディレクトリを含む現在のディレクトリ内にある、通常のファイルとディレクトリを標準出力に出力するコマンドラインを作成せよ。その際、“.”で始まるディレクトリも含めること。
- 例)


```
$>./count_files.sh | cat -e
42$
$>
```



失敗は学びの旅の一部です。

# Chapter VII

## Exercise 04: MAC

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

- あなたのコンピュータのMACアドレスを標準出力に出力するコマンドラインを作成せよ。その際、アドレスごとに改行すること。




`man ifconfig`



協力し合うことが、成功の鍵です。

## Chapter VIII

### Exercise 05: Can you create it ?

	Exercise 05
Can you create it ?	
提出するディレクトリ : <i>ex05/</i>	
提出するファイル : <code>"\?\$*'MaRViN'*\$?\\"</code>	
使用可能な関数 : None	

- “42” のみを含むファイルを作成せよ。
- そのファイルの名前は、以下のようになる。

```
"\?$*'MaRViN'*$?\\"
```

- 例)

```
$>ls -lRa *MaRV* | cat -e
-rw---xr-- 1 75355 32015 2 Oct 2 12:21 "\?$*'MaRViN'*$?\\"$
$>
```

## マイルストーン達成、引き続きがんばりましょう！

あなたは、この課題の最後の必須問題に到達しました。


次の課題に進むか、引き続きこの課題の追加問題に取り組むかは、あなた次第です。どちらの道を選んでも、いずれ役に立つ要素を学ぶことができます。

以下の要素を考慮して、選択してください。：

- 最初のExamはC言語に関する内容のため、あなたはすでに最初のC言語の課題を経験しているかもしれません。週末に行われるRushについても同様です。（Rushについてはすぐに学ぶことになります）
- Piscineの成績は、様々な要素によって評価されます。各課題のクリアはその一つの要素ですが、全課題の全体的な進捗もまた要素の一つです。最善の結果につながるように、賢明な選択をしてください。
- Piscineが終了するまでの間は、数日から数週間後に、同じ課題へ再び取り組むことが可能です。
- 他のPiscine受験生とある程度同じ進捗を保つことで、より円滑に学び合うことができます。

# Chapter IX

## Exercise 06: Skip

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


- `ls -l` コマンドの1行目から1行おきに、標準出力に出力するコマンドラインを作成せよ。



Gitを定期的にプッシュすること。

# Chapter X

## Exercise 07: r\_dwssap

	Exercise 07
	r_dwssap.sh
	提出するディレクトリ : <i>ex07/</i>
	提出するファイル : <b>r_dwssap.sh</b>
	使用可能な関数 : None

- `cat /etc/passwd` コマンドの出力を表示し、コメントを削除し、2行目から1行おきに、各ログインの文字列を逆順にし、アルファベット順の降順に並べ替え、FT\_LINE1とFT\_LINE2の間のログインのみが保持されたものを、“,”（引用符を除く）で区切り、出力は“.”で終わる標準出力に出力する関数を作成せよ。
- 例) 7から15行の間の結果は、以下のようになる。

```
$> ./r_dwssap.sh
sstq_, sorebrek_brk_, soibten_, sergtsop_, scodved_, rlaxcm_, rgmecived_, revreswodniw_, revressta_
.$>
```




必ず、指示通りの順序で取り組むこと。



隣のPiscine受験生に相談しましたか？

# Chapter XI

## Exercise 08: add\_chelou

	Exercise 08
	add_chelou.sh
	提出するディレクトリ : <i>ex08/</i>
	提出するファイル : <i>add_chelou.sh</i>
	使用可能な関数 : None

- '＼"?!'ベースの変数FT\_NBR1と、mrdocベースのFT\_NBR2から、数値を取得するコマンドラインを作成せよ。その際、gtaio luSnemfベースで両方の合計を標準出力に出力すること。

- 例1

```
FT_NBR1=\'＼"?!\'＼  
FT_NBR2=rcrdmddd
```

- 合計値

```
Salut
```

- 例2

```
FT_NBR1=\'＼"!＼"!＼"!＼"!＼"!＼"!＼"!＼"!＼"  
FT_NBR2=dcrmcmmoododmrrrmorcrcrmomo
```

- 合計値

```
Segmentation fault
```



他の受験生へ話しかけることを躊躇わずに、誰かに質問してみましょう。

# Chapter XII

## 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。