



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 : 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>11</b>
<b>X</b>	<b>Exercise 07 : r_dwssap</b>	<b>12</b>
<b>XI</b>	<b>Exercise 08 : add_chelou</b>	<b>13</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 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	

- 週の中に、アジェンダで金曜日の試験に登録することができますので忘れないでください。
- Exam00というプロジェクトにも登録する必要があります。
- 試験に登録できていることを確認してください（イベントとプロジェクトどちらにもです!）。
- 試験に登録できていることを確認したか、確認してください（イベントとプロジェクトも！両方必ずです!）。

# Chapter IV

## Exercise 01 : print\_groups

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

- 環境変数 FT\_USER に含まれるログインが、会員であるグループのリストを表示するコマンドラインを書きましょう。スペースなしでコンマで区切りましょう。

● 例)

- FT\_USER=noursの場合、結果は"god,root,admin,master,nours,bocal" (引用符なし)

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


- FT\_USER=daemonの場合、結果は"daemon,bin" (引用符なし)

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



man id

## 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 : <b>find_sh.sh</b>	
使用可能な関数 : None	


- 現在のディレクトリとそのサブディレクトリ全てから、".sh" (クエスチョンマークなし)で終わる名前の全てのファイルを探すコマンドを書きましょう。  
.shを消してファイル名だけを表示させましょう。
- 出力例)

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



# 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 : <b>count_files.sh</b>	
使用可能な関数 : None	

- 現在のディレクトリとそのサブファイルの全てから、普通のファイルとディレクトリを数えて表示するコマンドラインを書きましょう。 "."ディレクトリも含まれます。
- 出力例)

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

# 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 : <b>MAC.sh</b>
	使用可能な関数 : None


- あなたのコンピュータのMACアドレスが表示されるコマンドラインを書きましょう。各アドレスごとに改行しましょう。



`man ifconfig`

## 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"` だけを含むファイルを作成しましょう。
- そのファイルの名前は次のようになるでしょう :


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

- 例)

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

# 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 : <b>skip.sh</b>	
使用可能な関数 : None	

- `ls -l` コマンドの一行目から一行おきに表示するコマンドラインを書きましよう。

# Chapter X

## Exercise 07 : r\_dwssap

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


- cat /etc/passwd コマンドの出力を2行目から1行おきに表示し、コメントを削除し、各ログインを逆にし、アルファベット順の逆順に並べ替え、FT\_LINE1とFT\_LINE2 の間のログインのみを保持するコマンド行を作成する。尚、これらは "," (ダブルクォーテーションマークはなし) で区切り、出力は "."で終わらなければなりません。
- 例) 7から15行の間の結果は次のようになるでしょう

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



必ず、指示通りの順序で取り組みましょう。

## Exercise 08 : add\_chelou

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

- ハ"?!ベースの変数FT\_NBR1、およびmrdocベースのFT\_NBR2から、数値を取得するコマンドラインを書きましょう。 gtaio luSnemfベースで両方の合計を表示します。

- 例1

```
FT_NBR1=\ ' ? " \ " ' \
FT_NBR2=r cr d m d d d
```

- 合計値

Salut

- 例2

```
FT_NBR1="\\"!\\"!"!\\"!"!\\"!"!\\"!"!\\"!"
```

- 合計値

Segmentation fault