



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

I	Instructions	2
II	Foreword	4
III	Exercise 00 : Exam	5
IV	Exercise 01 : print_groups	6
V	Exercise 02 : find_sh	7
VI	Exercise 03 : count_files	8
VII	Exercise 04 : MAC	9
VIII	Exercise 05 : Can you create it ?	10
IX	Exercise 06 : Skip	11
X	Exercise 07 : r_dwssap	12
XI	Exercise 08 : add_chelou	13

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 : 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、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

Exercise 02 : find_sh




- 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 : <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を含めること。
- 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 : <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`

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'*$?\\"$
$>
```

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 : skip.sh	
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。

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、2行目から1行おきに標準出力に出力する関数を作成せよ。その際、これらは“,”（引用符を除く）で区切り、出力は“.”で終わること。
- 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。

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=dcrmcmmoododmrrrmorcmmomomo
```

- 合計値

```
Segmentation fault
```