



C Piscine

Shell 00

*Summary:* このドキュメントは、*C Piscine @ 42*の *Shell 00*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 : Z</b>	<b>6</b>
<b>IV</b>	<b>Exercise 01 : testShell00</b>	<b>7</b>
<b>V</b>	<b>Exercise 02 : Oh yeah, mooore...</b>	<b>8</b>
<b>VI</b>	<b>Exercise 03 : SSH me!</b>	<b>9</b>
<b>VII</b>	<b>Exercise 04 : midLS</b>	<b>10</b>
<b>VIII</b>	<b>Exercise 05 : GiT commit</b>	<b>11</b>
<b>IX</b>	<b>Exercise 06 : gitignore</b>	<b>12</b>
<b>X</b>	<b>Exercise 07 : diff</b>	<b>13</b>
<b>XI</b>	<b>Exercise 08 : clean</b>	<b>14</b>
<b>XII</b>	<b>Exercise 09 : Illusions, not tricks, Michael...</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 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 : Z


	Exercise 00
Only the best know how to display Z	
提出するディレクトリ : <i>ex00/</i>	
提出するファイル : <i>z</i>	
使用可能な関数 : <i>None</i>	

- `cat`というコマンドが使用されるたびに“Z”と改行を返す、`z`というファイルを作成せよ。

```
?>cat z
Z
?>
```

# Chapter IV

## Exercise 01 : testShell00

	Exercise 01
What are attributes anyway ?	
提出するディレクトリ : <i>ex01/</i>	
提出するファイル : <i>testShell00.tar</i>	
使用可能な関数 : None	

- 提出ディレクトリに *testShell00* というファイルを作成せよ。
- 出力が、以下のような方法を見つけること。(“total 1”の行は除く)

```
%> ls -l
total 1
-r--r-xr-x 1 XX XX 40 Jun 1 23:42 testShell00
%>
```

- 完了した場合は、以下のコマンドを実行し、提出ファイルを作成すること。  
`tar -cf testShell00.tar testShell00`



“XX”の代わりに、何か別のものが表示されていても問題ない。




ファイルのtimestamp上で、“時間”が“年”で表示されていても問題ない。



# Chapter V

## Exercise 02 : Oh yeah, mooore...

	Exercise 02
Oh yeah, mooore...	
提出するディレクトリ : <i>ex02/</i>	
提出するファイル : <i>exo2.tar</i>	
使用可能な関数 : None	

- ディレクトリで`ls -l`コマンドが使用される際、出力が以下のようなファイルとディレクトリを作成せよ。

```
%> ls -l
total XX
drwx--xr-x 2 XX XX XX Jun 1 20:47 test0
-rwx--xr-- 1 XX XX 4 Jun 1 21:46 test1
dr-x---r-- 2 XX XX XX Jun 1 22:45 test2
-r-----r-- 2 XX XX 1 Jun 1 23:44 test3
-rw-r-----x 1 XX XX 2 Jun 1 23:43 test4
-r-----r-- 2 XX XX 1 Jun 1 23:44 test5
lrwxrwxrwx 1 XX XX 5 Jun 1 22:20 test6 -> test0
%>
```

- 完了した場合は、以下のコマンドを実行し、提出ファイルを作成すること。  
`tar -cf exo2.tar *`




“XX” の代わりに、何か別のものが表示されていても問題ない。



ファイルのtimestamp上で、“時間” が“年” で表示されていても問題ない。

# Chapter VI

## Exercise 03 : SSH me!

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

- SSH鍵を発行すること。発行が完了した場合、
  - 公開鍵を *id\_rsa\_pub* というファイルの中にコピーして提出すること。
  - intranet上でSSH鍵をアップデートすること。これにより、gitのサーバに提出を行うことが可能となる。




提出ファイルの名前はランダムに選ばれていません。



公開鍵と秘密鍵の違いを理解すること。

# Chapter VII

## Exercise 04 : midLS

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

- *midLS* というファイルの中に、現在のディレクトリ内にある、すべてのファイルとディレクトリ（隠しファイルやドットで始まるファイルを除く - 二つのドットで始まる場合でも除くこと）をカンマで区切り、編集日順に一覧表示するコマンドラインを記入せよ。その際、ディレクトリの名前の後にスラッシュ文字が続いていることを確認すること。




求められていないことは、行わないこと。



RTFM!

# Chapter VIII

## Exercise 05 : GiT commit

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

- あなたのGitリポジトリにある、直近5つのコミットIDを出力するシェルスクリプトを作成せよ。

```
%> bash git_commit.sh | cat -e
baa23b54f0adb7bf42623d6d0a6ed4587e11412a$
2f52d74b1387fa80eea844969e8dc5483b531ac1$
905f53d98656771334f53f59bb984fc29774701f$
5ddc8474f4f15b3fcb72d08fcb333e19c3a27078$
e94d0b448c03ec633f16d84d63beaef9ae7e7be8$
%>
```


提出物の確認をする際は、我々の環境下で行われる。



RTFM!

# Chapter IX

## Exercise 06 : gitignore

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

- Gitのリポジトリに反映されない、すべてのファイルをリストアップする短いシェルスクリプトを作成せよ。例)

```
%> bash git_ignore.sh | cat -e
.DS_Store$
mywork.c~$
%>
```


提出物の確認をする際は、42の環境下で行われる。



RTFM!

# Chapter X

## Exercise 07 : diff

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

- 出力が以下のようなになるファイル**b**を作成せよ。

```
%>cat -e a
STARWAR$
Episode IV, A NEW HOPE It is a period of civil war.$
$
Rebel spaceships, striking from a hidden base, have won their first victory against the evil
Galactic Empire.$
During the battle, Rebel spies managed to steal secret plans to the Empire's ultimate weapon, the
DEATH STAR,$
an armored space station with enough power to destroy an entire planet.$
$
Pursued by the Empire's sinister agents, Princess Leia races home aboard her starship, custodian of
the stolen plans that can save her people and restore freedom to the galaxy...$
$
```


```
%>diff a b > sw.diff
```



man patch

# Chapter XI

## Exercise 08 : clean

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


- `clean`という名前のファイルに、現在のディレクトリとそのサブディレクトリの両方で、`~`で終わる名前、または`#`で始まり`#`で終わる名前のファイルを検索するためのコマンドラインを作成せよ。
- このコマンドラインは、見つかったファイルを出力したのちに削除する。
- コマンドは、1つのみ使用すること。“;” や “&&” などの黒魔術は禁止とする。



`man find`

## Chapter XII

### Exercise 09 : Illusions, not tricks, Michael...

	Exercise 09
Illusions, not tricks, Michael...	
提出するディレクトリ : <code>ex09/</code>	
提出するファイル : <code>ft_magic</code>	
使用可能な関数 : <code>None</code>	

- 42バイト目から“42”という文字列が始まる、`42 file`というファイルタイプを検出する、`ft_magic`というマジックファイルを作成せよ。



`man file`