



Piscina C

Shell 01

Sommario: Questo documento tratta il modulo Shell-01 della Piscina C @ 42.

Indice

I	Istruzioni	2
II	Preambolo	3
III	Esercizio 00 : Exam	4
IV	Esercizio 01 : print_groups	5
V	Esercizio 02 : find_sh	6
VI	Esercizio 03 : count_files	7
VII	Esercizio 04 : MAC	8
VIII	Exercise 05 : Riesci a crearlo ?	9
IX	Exercise 06 : Skip	10
X	Esercizio 07 : r_dwssap	11
XI	Esetcizio 08 : add_chelou	12

Capitolo I

Istruzioni

- Fate riferimento solo a questa pagina: non fidatevi delle dicerie.
- Questo documento può subire variazioni prima della scadenza per la presentazione, verificate quindi il suo contenuto prima di presentare gli esercizi.
- Gli esercizi sono presentati seguendo un ordine di difficoltà crescente. Ai fini della valutazione **NON** si prendono in considerazione gli esercizi se i precedenti non sono stati completati correttamente.
- Controllate i permessi dei vostri file e delle vostre cartelle.
- Dovete seguire le procedure di presentazione per tutti gli esercizi.
- I vostri esercizi saranno controllati e valutati dai vostri compagni di corso.
- I vostri esercizi saranno soggetti, oltre alla valutazione tra pari, al controllo e alla valutazione da parte di un programma chiamato Moulinette.
- Moulinette sarà estremamente meticolosa e severa nel valutare il vostro lavoro. Essendo il suo un processo automatico senza possibilità di ricorso, assicuratevi di essere il più precisi possibile al fine di evitare brutte sorprese.
- Gli esercizi del modulo Shell devono essere eseguibili tramite `/bin/sh`.
- **NON** sarà tollerato **ALCUN** file aggiuntivo nelle cartelle presentate oltre a quelli specificati in questo documento.
- Dubbi o domande? Chiedi a chi si trova alla tua destra, altrimenti a chi si trova alla tua sinistra.
- I vostri manuali si chiamano **Google / man / Internet /**
- Date un'occhiata alla sezione Piscina C del forum dell'Intranet.
- Prestate attenzione agli esempi proposti, in quanto potrebbero mostrare dettagli non esplicitamente presentati nel documento...

Capitolo II

Preambolo

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.

Capitolo III


Esercizio 00 : Exam

	Esercizio : 00
Exam	

- Durante questa settimana avrete la possibilità di iscrivervi all'evento Esame di venerdì presente in agenda.
- Dovete anche iscrivervi al progetto Exam00.
- Assicuratevi di aver effettuato l'iscrizione all'Esame (evento E progetto !).
- Assicuratevi di esservi assicurati di aver effettuato l'iscrizione all'Esame (evento E progetto ! Entrambi !).

Capitolo IV

Esercizio 01 : print_groups

	Esercizio 01
	print_groups.sh
	Cartella per la consegna : <i>ex01/</i>
	File da consegnare : print_groups.sh
	Funzioni permesse : Nessuna

- Scrivere un comando di linea che stampi a video la lista dei gruppi ai quali il login, contenuto nella variabile d'ambiente FT_USER, appartiene. I risultati dovranno essere presentati separati da virgole e privi di spazi.
- Esempi :
 - Per FT_USER=nours, Il risultato deve essere "god,root,admin,master,nours,bocal" (senza virgolette)

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

- Per FT_USER=daemon, il risultato deve essere "daemon,bin" (senza virgolette)

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



man id


Esercizio 02 : find_sh



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

# Capitolo VI

## Esercizio 03 : count\_files

|                                                                                   |                                            |
|-----------------------------------------------------------------------------------|--------------------------------------------|
|  | Esercizio 03                               |
|                                                                                   | count_files.sh                             |
|                                                                                   | Cartella per la consegna : <i>ex03/</i>    |
|                                                                                   | File da consegnare : <b>count_files.sh</b> |
|                                                                                   | Funzioni permesse : <b>Nessuna</b>         |

- Scrivere un comando di linea che stampi a video il risultato di un conteggio di tutti i file e cartelle regolari presenti nella cartella attuale e in tutte le sotto-cartelle. Nel conteggio deve essere inclusa ".", la cartella in cui viene eseguito il comando.
- Esempio di output :

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



# Capitolo VII

## Esercizio 04 : MAC

|                                                                                   |              |
|-----------------------------------------------------------------------------------|--------------|
|  | Esercizio 04 |
|                                                                                   | MAC.sh       |
| Cartella per la consegna : <i>ex04/</i>                                           |              |
| File da consegnare : <b>MAC.sh</b>                                                |              |
| Funzioni permesse : <b>Nessuna</b>                                                |              |


- Scrivere un comando di linea che stampi a video gli indirizzi MAC della vostra macchina. Ogni indirizzo deve essere seguito da una nuova linea.



`man ifconfig`

# Capitolo VIII

## Exercise 05 : Riesci a crearlo ?

|                                                                                   |              |
|-----------------------------------------------------------------------------------|--------------|
|  | Esercizio 05 |
| Riesci a crearlo ?                                                                |              |
| Cartella per la consegna : <i>ex05/</i>                                           |              |
| File da consegnare : "\?*\$*'MaRViN'*\$?\\"                                       |              |
| Funzioni permesse : Nessuna                                                       |              |

- Creare un file che contenga SOLAMENTE "42"(senza virgolette).
- Il nome del file deve essere :


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

- Esempio :

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

# Capitolo IX


## Exercise 06 : Skip

|                                                                                   |              |
|-----------------------------------------------------------------------------------|--------------|
|  | Esercizio 06 |
| skip.sh                                                                           |              |
| Cartella per la consegna : <i>ex06/</i>                                           |              |
| File da consegnare : <b>skip.sh</b>                                               |              |
| Funzioni permesse : <b>Nessuna</b>                                                |              |

- Scrivere un comando di linea che stampi a video una linea ogni due, partendo dalla prima, del risultato di un comando `ls -l`.

# Capitolo X

## Esercizio 07 : r\_dwssap

|                                                                                   |                                         |
|-----------------------------------------------------------------------------------|-----------------------------------------|
|  | Esercizio 07                            |
|                                                                                   | r_dwssap.sh                             |
|                                                                                   | Cartella per la consegna : <i>ex07/</i> |
|                                                                                   | File da consegnare : <b>r_dwssap.sh</b> |
|                                                                                   | Funzioni permesse : <b>Nessuna</b>      |

- Scrivere un comando di linea che stampi a video l'output del comando `cat /etc/passwd` dopo aver : rimosso i commenti; mantenuto una linea ogni due a partire dalla seconda; invertito ogni login; ordinato il tutto in ordine alfabetico inverso; mantenuto solo i login tra FT\_LINE1 e FT\_LINE2 inclusa separati da ", "(senza virgolette). L'output deve terminare con ".".
- Esempio: tra la linea 7 e la linea 15, l'output sarà qualcosa di simile:


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



Rispettate rigorosamente l'ordine delle istruzioni.

# Capitolo XI

## Esercizio 08 : add\_chelou

|                                                                                   |                                           |
|-----------------------------------------------------------------------------------|-------------------------------------------|
|  | Esercizio 08                              |
|                                                                                   | add_chelou.sh                             |
|                                                                                   | Cartella per la consegna : <i>ex08/</i>   |
|                                                                                   | File da consegnare : <b>add_chelou.sh</b> |
|                                                                                   | Funzioni permesse : <b>Nessuna</b>        |

- Scrivere un comando di linea che dati i numeri contenuti nelle variabili FT\_NBR1( base '\"?!) e FT\_NBR2( base mrdoc) ne effettui la somma e ne mostri il risultato in base gtaio luSnemf.

- Esempio 1:

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

- La somma é :

```
Salut
```

- Esempio 2 :

```
FT_NBR1="\\"!\\\\"!\\\\"!\\\\"!\\\\"!\\\\"!\\\\"!\\\\"
FT_NBR2=dcrcmcmoododmrrrmorcrcrmomo
```

- La somma é :

```
Segmentation fault
```