



# Piscine C

## Shell 00

*Summary: Ce document est le sujet du module Shell 00 de la piscine C de 42.*

*Version: 4.1*

# Contents

<b>I</b>	<b>Consignes</b>	<b>2</b>
<b>II</b>	<b>Préambule</b>	<b>3</b>
<b>III</b>	<b>Exercice 00 : Z</b>	<b>4</b>
<b>IV</b>	<b>Exercice 01 : testShell00</b>	<b>5</b>
<b>V</b>	<b>Exercice 02 : Owi, encore...</b>	<b>6</b>
<b>VI</b>	<b>Exercice 03 : Connecte-moi!</b>	<b>7</b>
<b>VII</b>	<b>Exercice 04 : midLS</b>	<b>8</b>
<b>VIII</b>	<b>Exercice 05 : GiT commit</b>	<b>9</b>
<b>IX</b>	<b>Exercice 06 : gitignore</b>	<b>10</b>
<b>X</b>	<b>Exercice 07 : diff</b>	<b>11</b>
<b>XI</b>	<b>Exercice 08 : clean</b>	<b>12</b>
<b>XII</b>	<b>Exercice 09 : Illusions, not tricks, Michael...</b>	<b>13</b>
<b>XIII</b>	<b>Rendu et peer-evaluation</b>	<b>14</b>

# Chapter I

## Consignes

- Seule cette page servira de référence : ne vous fiez pas aux bruits de couloir.
- Relisez bien le sujet avant de rendre vos exercices. A tout moment le sujet peut changer.
- Les exercices sont très précisément ordonnés du plus simple au plus complexe. En aucun cas nous ne porterons attention ni ne prendrons en compte un exercice complexe si un exercice plus simple n'est pas parfaitement réussi.
- Attention aux droits de vos fichiers et de vos répertoires.
- Vous devez suivre la procédure de rendu pour tous vos exercices.
- Vos exercices seront corrigés par vos camarades de piscine.
- En plus de vos camarades, vous serez corrigés par un programme appelé la Moulinette.
- La Moulinette est très stricte dans sa notation. Elle est totalement automatisée. Il est impossible de discuter de sa note avec elle. Soyez donc rigoureux !
- Les exercices shell doivent s'exécuter avec `/bin/sh`.
- Vous ne devez laisser dans votre répertoire aucun autre fichier que ceux explicitement spécifiés par les énoncés des exercices.
- Vous avez une question ? Demandez à votre voisin de droite. Sinon, essayez avec votre voisin de gauche.
- Votre manuel de référence s'appelle `Google / man / Internet / ....`
- Lisez attentivement les exemples. Ils pourraient bien requérir des choses qui ne sont pas autrement précisées dans le sujet...

# Chapter II

## Préambule

Voici les paroles du générique de Nicky Larson:

Une ombre file dans la nuit  
C'est un assassin qui s'enfuit  
Et comme un démon il sourit  
Son crime restera impuni  
Une voiture qui surgit  
Un coup de frein, des pneus qui crient  
Un coup de feu qui retentit  
La justice s'appelle Nicky

[Refrain]  
Dans la chaleur  
De la nuit  
Le mal est toujours puni  
Aucun danger ne l'impressionne  
Les coups durs il les affectionne  
Et la justice le passionne  
Nicky Larson ne craint personne  
Lorsque les coups de feu résonnent  
Comme un éclair il tourbillonne  
Surtout si la fille est mignonne  
Nicky Larson ne craint personne


Comme un chasseur il suit sa proie  
Pour que la justice et le droit  
Triomphent, il est prêt à donner  
Toute sa vie sans hésiter  
Quand sa silhouette apparaît  
Les méchants se mettent à trembler  
Ils savent qu'ils ne pourront jamais  
Echapper à ce justicier

[Refrain]

Ce sujet n'a, malheureusement, rien à voir avec Nicky Larson.

# Chapter III

## Exercice 00 : Z


	Exercise 00
Les vrais savent afficher un Z	
Turn-in directory : <i>ex00/</i>	
Files to turn in : <b>z</b>	
Allowed functions : <b>None</b>	

- Créez un fichier **z** qui, lorsque l'on fait un **cat** dessus, affiche "Z" suivi d'un retour à la ligne.

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

# Chapter IV

## Exercice 01 : testShell00

	Exercice 01
C'est quoi des attributs ?	
Turn-in directory : <i>ex01/</i>	
Files to turn in : <b>testShell00.tar</b>	
Allowed functions : None	

- Créer le fichier **testShell00** dans votre répertoire de rendu.
- Vous devrez faire en sorte que:

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

- Une fois l'exercice résolu, vous exécuterez la commande **tar -cf testShell00.tar testShell00** pour créer le fichier à rendre.




Les "XX" ne seront pas pris en compte.



Une année est une réponse acceptée à la place de 1'heure.

# Chapter V

## Exercice 02 : Owi, encore...

	Exercise 02
Owi, encore...	
Turn-in directory : <i>ex02/</i>	
Files to turn in : <b>exo2.tar</b>	
Allowed functions : None	

- Créer tous ces fichiers et répertoires. Faire le nécessaire pour que l’affichage d’un `ls -l` dans votre répertoire ressemble à cela :

```
%> 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
%>
```

- Une fois l’exercice résolu, vous exécuterez la commande `tar -cf exo2.tar *` pour créer le fichier à rendre.




Les "XX" ne seront pas pris en compte.



Une année est une réponse acceptée à la place de 1’heure.

# Chapter VI

## Exercice 03 : Connecte-moi!

	Exercice 03
SSH key	
Turn-in directory : <i>ex03/</i>	
Files to turn in : <i>id_rsa_pub</i>	
Allowed functions : <i>None</i>	

- Créer vos propre clés SSH. Ensuite :
  - Ajoutez votre clé SSH publique à votre rendu dans un fichier *id\_rsa\_pub*.
  - Mettez à jour votre clé SSH publique sur l'intra pour vous autoriser à téléverser vos projets sur notre serveur git.



Le nom du fichier n'a pas été choisi au hasard.




Ces commandes vous seront utiles pendant toute votre piscine, ne les oubliez pas !



# Chapter VII

## Exercice 04 : midLS

	Exercise 04
midLS	
Turn-in directory : <i>ex04/</i>	
Files to turn in : <b>midLS</b>	
Allowed functions : <b>None</b>	

- Placez dans un fichier **midLS** la ligne de commande à taper pour lister les fichiers et les répertoires du répertoire courant, mais pas les fichiers cachés, ni "." ni ".." (rien ne commençant par un point), séparés par des virgules, triés par date de modification et de manière à ce que les répertoires soient suivis d'un caractère slash.




Ce qui n'est pas demandé n'est pas à faire !



RTFM !

# Chapter VIII

## Exercise 05 : GiT commit

	Exercise 05
GiT commit?	
Turn-in directory : <i>ex05/</i>	
Files to turn in : <i>git_commit.sh</i>	
Allowed functions : None	

- Créez un script shell qui renvoie les id des 5 derniers commit de votre dépôt git.

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


Votre script sera teste dans notre environnement.



RTFM!

# Chapter IX

## Exercise 06 : gitignore

	Exercise 06
GiT	
Turn-in directory : <i>ex06/</i>	
Files to turn in : <code>git_ignore.sh</code>	
Allowed functions : None	

- Écrivez un court script shell qui renverra la liste des fichiers existants ignorés par votre dépôt git **actuellement présents** dans votre dépôt local. Exemple:

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


Votre script sera teste dans notre environnement.



RTFM!

# Chapter X

## Exercice 07 : diff

	Exercise 07
Turn-in directory : <i>ex07/</i>	
Files to turn in : <b>b</b>	
Allowed functions : <b>None</b>	

- Créer le fichier **b** tel que :

```
%>cat -e a
STARWARSS$
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

## Exercice 08 : clean

	Exercise 08
Turn-in directory : <i>ex08/</i>	
Files to turn in : <b>clean</b>	
Allowed functions : <b>None</b>	


- Placez dans un fichier **clean** une ligne de commande qui va rechercher, à partir du répertoire courant et dans tous ses sous-répertoires, les fichiers dont le nom se termine par `~`, ou commence et se termine par `#`.
- La ligne de commande affichera et effacera les fichiers trouvés.
- Une seule commande est autorisée : pas de `';` ou de `'&&'` ou autre.



`man find`

## Chapter XII

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

	Exercise 09
Illusions, not tricks, Michael...	
Turn-in directory : <i>ex09/</i>	
Files to turn in : <b>ft_magic</b>	
Allowed functions : <b>None</b>	

- Créer un fichier magique nommé **ft\_magic** qui devra être correctement formaté pour détecter les fichiers de type **42 file** et constitués de la chaîne "42" à la 42ème octet.



man file

# Chapter XIII

## Rendu et peer-evaluation

Rendez votre travail sur votre dépôt `Git` comme d'habitude. Seul le travail présent sur votre dépôt sera évalué en soutenance. Vérifiez bien les noms de vos dossiers et de vos fichiers afin que ces derniers soient conformes aux demandes du sujet.

Vu que votre travail ne sera pas évalué par un programme, organisez vos fichiers comme bon vous semble du moment que vous rendez les fichiers obligatoires et respectez les consignes du sujet.



`Vous ne devez rendre uniquement les fichiers demandés par le sujet de ce projet.`