



Piscine C

Shell 00

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

Version: 4.1

Contents

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

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 : z	
Allowed functions : None	

- 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 : testShell00.tar	
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 : exo2.tar	
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 : midLS	
Allowed functions : None	

- 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	
Allowed functions : None	

- 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 : clean	
Allowed functions : None	


- 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 : <code>ft_magic</code>	
Allowed functions : <code>None</code>	

- 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.`



Piscine C

Shell 01

Résumé: Ce document est le sujet du module Shell 01 de la piscine C de 42.

Version:

Table des matières

I	Consignes	2
II	Préambule	3
III	Exercice 00 : Exam machine	4
IV	Exercice 01 : print__groups	5
V	Exercice 02 : find__sh	6
VI	Exercice 03 : count__files	7
VII	Exercice 04 : MAC	8
VIII	Exercice 05 : Can you create it ?	9
IX	Exercice 06 : Skip	10
X	Exercice 07 : r__dwssap	11
XI	Exercice 08 : add__chelou	12
XII	Rendu et peer-evaluation	13

Chapitre 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 /`
- Pensez à discuter sur le forum Piscine de votre Intra, ainsi que sur le slack de votre Piscine !
- Lisez attentivement les exemples. Ils pourraient bien requérir des choses qui ne sont pas autrement précisées dans le sujet...

Chapitre II

Préambule

Voici ce que Wikipédia a à dire à propos de la loutre :

La Loutre d'Europe ou Loutre européenne (*Lutra lutra*), souvent qualifiée de loutre commune dans les pays d'Europe où elle est présente, est un mammifère carnivore semi-aquatique et principalement nocturne, de la famille des Mustélidés (sous-famille Lutrinés). Elle est l'une des trois espèces de loutres se rattachant au genre *Lutra*. En France, on ne trouve que cette seule espèce de loutre.

Sa hauteur est d'environ 30 cm au garrot. Son pelage, brun foncé, est composé de deux couches : le poil de bourre, court, très fin, dense et laineux ; et le poil de jarre, long, lisse, brillant et imperméable.

Excellente nageuse, elle dispose de pattes palmées, d'un corps allongé (60 à 80 cm en moyenne, auquel il faut ajouter une queue épaisse à la base et s'effilant vers l'extrémité de 30 à 40 cm de longueur), pour un poids allant de 5 à 15 kg.


Elle vit au bord des cours d'eau (ruisseaux, rivières et même fleuves), jusqu'à une altitude de 1300 m, dans les marais et parfois sur les côtes marines. Elle est habituellement solitaire, occupant un territoire de 5 à 15 km de rives le long d'un cours d'eau (parfois davantage) ou de 20 à 30 km² en zone de marais. Elle emprunte régulièrement les mêmes passages sur la berge pour se mettre à l'eau : les "coulées". Lorsqu'elle sort de l'eau, elle se roule dans l'herbe pour essuyer sa fourrure, sur des zones reconnaissables à l'herbe couchée et appelées "places de ressui".

Elle fait sa tanière (qu'on appelle une "catiche", de l'ancien français "se catir" = se blottir, se cacher) entre les racines des arbres des berges des cours d'eau ou dans d'autres cavités (cavité rocheuse, tronc creux, terrier d'une autre espèce). La catiche contient souvent une entrée plus ou moins dissimulée au-dessous du niveau d'eau et un conduit d'aération.

C'est mignon, une loutre.

Chapitre III


Exercice 00 : Exam machine

	Exercice : 00
	Exam machine

- Dans la semaine, vous pourrez vous inscrire à l'exam de vendredi dans l'Agenda, pensez-y.
- Vous devez également vous inscrire au projet Exam00.
- Vérifiez que vous êtes bien inscrit à l'exam machine de vendredi (à la fois à l'événement ET au projet!).
- Vérifiez que vous avez vérifié que vous étiez inscrit (à l'événement ET au projet ! Oui, les deux!).

Chapitre IV

Exercice 01 : print_groups

	Exercice : 01
	print_groups.sh
	Dossier de rendu : ex01/
	Fichiers à rendre : print_groups.sh
	Fonctions Autorisées : Aucune

- Écrire une ligne de commande qui détermine et affiche la liste des groupes dans lesquels est membre le login spécifié dans la variable d'environnement FT_USER, séparés par des virgules sans espaces.
- Exemples :
 - avec FT_USER=nours, le résultat est "god,root,admin,master,nours,bocal" (sans les guillemets)

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

- avec FT_USER=daemon, le résultat est "daemon,bin" (sans les guillemets)


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



man id

Chapitre V

Exercice 02 : find_sh


	Exercice : 02
find_sh.sh	
Dossier de rendu : <i>ex02/</i>	
Fichiers à rendre : find_sh.sh	
Fonctions Autorisées : Aucune	

- Écrire une ligne de commande qui cherche dans le répertoire courant et dans tous ses sous-répertoires tous les fichiers dont le nom se termine par ".sh" (sans les guillemets) et n'affiche que leurs noms, sans le .sh.
- Exemple de sortie :

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

Chapitre VI

Exercice 03 : count_files


	Exercice : 03
	count_files.sh
Dossier de rendu : <i>ex03/</i>	
Fichiers à rendre : count_files.sh	
Fonctions Autorisées : Aucune	

- Écrire une ligne de commande qui affiche le nombre de fichiers réguliers et de répertoires dans le répertoire courant et tous ses sous-répertoires, y compris le "." du répertoire de départ.
- Exemple de sortie :

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

Chapitre VII

Exercice 04 : MAC

	Exercice : 04
MAC.sh	
Dossier de rendu : <i>ex04/</i>	
Fichiers à rendre : MAC.sh	
Fonctions Autorisées : Aucune	


- Écrire une ligne de commande qui affiche les adresses MAC de votre machine. Chaque adresse sera suivi d'un retour à la ligne.



`man ifconfig`

Chapitre VIII

Exercice 05 : Can you create it ?

	Exercice : 05
Can you create it ?	
Dossier de rendu : <i>ex05/</i>	
Fichiers à rendre : <code>"\?\${}*'MaRViN'*\$?\\"</code>	
Fonctions Autorisées : Aucune	

- Créer un fichier contenant uniquement "42" et **RIEN** d'autre.
- Il se nommera :


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

- Exemple :

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

Chapitre IX


Exercice 06 : Skip

	Exercice : 06
skip.sh	
Dossier de rendu : <i>ex06/</i>	
Fichiers à rendre : skip.sh	
Fonctions Autorisées : Aucune	

- Écrire une ligne de commande et qui affiche un `ls -l` une ligne sur deux, à partir de la première.

Chapitre X

Exercice 07 : r_dwssap

	Exercice : 07
	r_dwssap.sh
	Dossier de rendu : <i>ex07/</i>
	Fichiers à rendre : r_dwssap.sh
	Fonctions Autorisées : Aucune

- Écrire une ligne de commande qui affiche la sortie d'un `cat /etc/passwd`, en retirant les commentaires, une ligne sur deux en partant de la seconde en inversant chaque login et en triant par ordre alphabétique inversé, en ne conservant que les logins compris entre FT_LINE1 et FT_LINE2 inclus, séparés par des ", " (sans les guillemets), et terminés par un ".".
- Exemple : Entre les lignes 7 et 15, le résultat sera quelque chose du genre :

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



Respectez strictement l'ordre de l'énoncé.

Exercice 08 : add_chelou



- Exemple 1 :

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

- La somme est :

Salut

- Exemple 2 :

[illegible]

- La somme est :

```
Segmentation fault
```

Chapitre XII

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.`