

# INF1070 Utilisation et administration des systèmes informatiques

## Examen Intra — Automne 2018

### Samedi 27 octobre — Durée 3 heures

### Jean Privat & Alexandre Blondin Massé

Aucun document ni appareil électronique n'est autorisé. Répondez et rendez seulement le formulaire (dernière page). L'annexe est détachable.

Les questions faisant apparaître le symbole ♣ peuvent présenter plusieurs bonnes réponses. Les autres ont une unique bonne réponse.

Chacune des 20 questions vaut 5 points. Des points négatifs pourront être affectés à de très mauvaises réponses.

## 1 Questions générales

**Question 1 ♣** Lesquels des énoncés suivants sont vrais ?

- |   |  |   |
|---|--|---|
| <input type="checkbox"/> A 12Mio > 12Mo         | <input type="checkbox"/> D 1Mo/h > 1Mb/s   | <input type="checkbox"/> G 1ko = 8kb      |
| <input type="checkbox"/> B 512Go > 256To        | <input type="checkbox"/> E 10ko = 10240kio | <input type="checkbox"/> H 10Mo > 100kio  |
| <input type="checkbox"/> C 1Mo > 1To > 1ko > 1o | <input type="checkbox"/> F 5000kb > 2Mo    | <input type="checkbox"/> I 1Gio = 1024Mio |

Unités : b = bit, h = heure, o = octet, s = seconde.

Préfixes SI : G = giga, Gi = gibibyte, k = kilo, ki = kibi, M = méga, Mi = mébibyte, T = téra, Ti = tébibyte.

**Question 2** Qu'est-ce que Vim ?

- |   |   |
|---|---|
| <input type="checkbox"/> A Un format de fichier d'image vectorielle | <input type="checkbox"/> F Un système d'exploitation d'Oracle               |
| <input type="checkbox"/> B Un explorateur graphique de fichiers     | <input type="checkbox"/> G Une norme de développement de la Fondation Linux |
| <input type="checkbox"/> C L'ancêtre des systèmes UNIX              | <input type="checkbox"/> H Le vrai nom du noyau Linux                       |
| <input type="checkbox"/> D Une distribution Linux                   | <input type="checkbox"/> I Le prénom du développeur principal de Bash       |
| <input type="checkbox"/> E Un éditeur de texte en console           | <input type="checkbox"/> J Une norme de l'IEEE                              |

**Question 3 ♣** Lesquelles des affirmations suivantes sont vraies ?

- |   |
|---|
| <input type="checkbox"/> A La commande <code>gzip</code> permet aussi de compresser l'entrée standard   |
| <input type="checkbox"/> B Un fichier compressé avec <code>gzip</code> , puis décompressé avec <code>gunzip</code> est toujours identique au fichier original |
| <input type="checkbox"/> C Tout format de fichier d'image est compressé   |
| <input type="checkbox"/> D La commande <code>tar</code> permet d'archiver un ensemble de fichiers   |
| <input type="checkbox"/> E Un fichier compressé est toujours de plus petite taille que le fichier original avant compression                                  |

## 2 Fichiers et répertoires

**Question 4** Laquelle parmi les commandes suivantes permet de remonter au répertoire parent ?

- |   |  |   |  |   |  |
|---|--|---|--|---|--|
| <input type="checkbox"/> A <code>dir .</code> | <input type="checkbox"/> C <code>mv .</code> | <input type="checkbox"/> E <code>dir /</code> | <input type="checkbox"/> G <code>cd .</code> | <input type="checkbox"/> I <code>cd ~</code>  | <input type="checkbox"/> K <code>dir ..</code> |
| <input type="checkbox"/> B <code>mv ..</code> | <input type="checkbox"/> D <code>mv /</code> | <input type="checkbox"/> F <code>mv ~</code>  | <input type="checkbox"/> H <code>cd /</code> | <input type="checkbox"/> J <code>cd ..</code> | <input type="checkbox"/> L <code>dir ~</code>  |

**Question 5 ♣** Le fichier `important.top.secret.txt` est très important. Parmi les motifs de fichiers suivants (glob), lesquelles reconnaissent ce nom de fichier ?

- |  |   |   |  |
|--|---|---|--|
| <input type="checkbox"/> A <code>*.t[po]*</code>           | <input type="checkbox"/> D <code>*.*.*</code>       | <input type="checkbox"/> G <code>*.txt</code> | <input type="checkbox"/> J <code>*t*t*t*t*t*t*t*t</code> |
| <input type="checkbox"/> B <code>*[bdfghjklquvwyz]*</code> | <input type="checkbox"/> E <code>important.*</code> | <input type="checkbox"/> H <code>[a-z]</code> | <input type="checkbox"/> K <code>*???????</code>         |
| <input type="checkbox"/> C <code>*t*x?</code>              | <input type="checkbox"/> F <code>?i*</code>         | <input type="checkbox"/> I <code>*.top</code> | <input type="checkbox"/> L <code>?</code>                |

### 3 Tubes et redirections

La commande « `ls -li` » affiche :

```
total 4
18885716 crw-r--r--    1 privat  privat    89,    1 Oct 19 10:48 ananas
18885741 -rw-r--r--    1 privat  privat   1038 Oct 19 10:48 banane
18885723 brw-r--r--    1 privat  privat    45,   15 Oct 19 10:48 bleuete
18885731 -rw-r--r--    1 privat  privat    91 Oct 19 10:48 fraise
19408737 drwxr-xr-x    2 privat  privat   4096 Oct 19 10:48 grenade
18885742 lrwxrwxrwx    1 privat  privat    9 Oct 19 10:48 melon d'eau -> pastèque
18885743 -rw-r--r--    1 privat  privat  97047 Oct 19 10:48 pastèque
18885745 -rw-r--r--    1 privat  privat   141 Oct 19 10:48 poire
18885747 -rw-r--r--    1 privat  privat   150 Oct 19 10:48 pomme
```

**Question 6** Que fait la commande « `cat banane >> poire` » ?

- |                            |  |                            |  |
|----------------------------|--|----------------------------|--|
| <input type="checkbox"/> A | Affiche le contenu du fichier <code>poire</code> à l'écran                     | <input type="checkbox"/> I | Écrit le mot « <code>poire</code> » à l'écran                                  |
| <input type="checkbox"/> B | Écrit le mot « <code>banane</code> » à la fin du fichier <code>poire</code>    | <input type="checkbox"/> J | Copie le fichier <code>banane</code> vers le fichier <code>poire</code>        |
| <input type="checkbox"/> C | Écrit les mots « <code>poire</code> » et « <code>banane</code> » à l'écran     | <input type="checkbox"/> K | Renomme le fichier <code>banane</code> en « <code>poire</code> »               |
| <input type="checkbox"/> D | Écrit le mot « <code>banane</code> » à l'écran                                 | <input type="checkbox"/> L | Renomme le fichier <code>poire</code> en « <code>banane</code> »               |
| <input type="checkbox"/> E | Écrit le mot « <code>poire</code> » à la fin du fichier <code>banane</code>    | <input type="checkbox"/> M | Affiche le contenu des deux fichiers à l'écran                                 |
| <input type="checkbox"/> F | Écrit le mot « <code>banane</code> » à la fin du fichier <code>poire</code>    | <input type="checkbox"/> N | Écrit le mot « <code>poire</code> » à la fin du fichier <code>banane</code>    |
| <input type="checkbox"/> G | Copie le fichier <code>poire</code> vers le fichier <code>banane</code>        | <input type="checkbox"/> O | Affiche le contenu du fichier <code>banane</code> à l'écran                    |
| <input type="checkbox"/> H | Ajoute le contenu du fichier <code>banane</code> au fichier <code>poire</code> | <input type="checkbox"/> P | Ajoute le contenu du fichier <code>poire</code> au fichier <code>banane</code> |

**Question 7** ♣ Après la commande précédente, qu'est-ce qui change lorsque l'on exécute à nouveau « `ls -li` » ?

- |                            |  |                            |   |
|----------------------------|--|----------------------------|---|
| <input type="checkbox"/> A | Le nom du fichier <code>banane</code> a changé.          | <input type="checkbox"/> I | Le groupe propriétaire de <code>poire</code> a changé.      |
| <input type="checkbox"/> B | Le nom du fichier <code>poire</code> a changé.           | <input type="checkbox"/> J | Le groupe propriétaire de <code>banane</code> a changé.     |
| <input type="checkbox"/> C | Le nombre de liens durs de <code>poire</code> a changé.  | <input type="checkbox"/> K | L'utilisateur propriétaire de <code>poire</code> a changé.  |
| <input type="checkbox"/> D | Le numéro d'inœud de <code>poire</code> a changé.        | <input type="checkbox"/> L | Les permissions de <code>poire</code> ont changé.           |
| <input type="checkbox"/> E | La taille de <code>poire</code> a changé.                | <input type="checkbox"/> M | Le numéro d'inœud de <code>banane</code> a changé.          |
| <input type="checkbox"/> F | La date de modification de <code>banane</code> a changé. | <input type="checkbox"/> N | La taille de <code>banane</code> a changé.                  |
| <input type="checkbox"/> G | Le nombre de liens durs de <code>banane</code> a changé. | <input type="checkbox"/> O | La date de modification de <code>poire</code> a changé.     |
| <input type="checkbox"/> H | Les permissions de <code>banane</code> ont changé.       | <input type="checkbox"/> P | L'utilisateur propriétaire de <code>banane</code> a changé. |

**Question 8** Quel est le résultat de la commande « `ls | tail -n 2 | head -n 1` » ?

Attention, c'est « `ls` », pas « `ls -li` » ici.

- |                            |       |                            |           |                            |        |                            |            |
|----------------------------|-------|----------------------------|-----------|----------------------------|--------|----------------------------|------------|
| <input type="checkbox"/> A | total | <input type="checkbox"/> D | poire     | <input type="checkbox"/> G | 10:48  | <input type="checkbox"/> J | ananas     |
| <input type="checkbox"/> B | 89, 1 | <input type="checkbox"/> E | pomme     | <input type="checkbox"/> H | oct    | <input type="checkbox"/> K | -rw-r--r-- |
| <input type="checkbox"/> C | 1     | <input type="checkbox"/> F | head -n 1 | <input type="checkbox"/> I | privat | <input type="checkbox"/> L | 141        |

**Question 9** Même question pour « `ls -li | cut -c 7 | sort | uniq -c | sort -n | head -n 1` »

- |                            |           |                            |         |                            |     |                            |     |
|----------------------------|-----------|----------------------------|---------|----------------------------|-----|----------------------------|-----|
| <input type="checkbox"/> A | head -n 1 | <input type="checkbox"/> D | total   | <input type="checkbox"/> G | 1 1 | <input type="checkbox"/> J | 1 4 |
| <input type="checkbox"/> B | ananas    | <input type="checkbox"/> E | total 4 | <input type="checkbox"/> H | 89  | <input type="checkbox"/> K | 2 3 |
| <input type="checkbox"/> C | 1         | <input type="checkbox"/> F | 18      | <input type="checkbox"/> I | 38  | <input type="checkbox"/> L | 16  |

## 4 Chercher dans des répertoires

Dans un répertoire initialement vide, on entre les commandes suivantes

```
$ mkdir -p a/b/c
$ mkdir -p a/a/b
$ mkdir -p b/a
$ echo "c" > a/c
$ echo "b" > c
```

**Question 10 ♣** Quelles lignes, parmi les suivantes, seront affichées par la commande « `find .` » ?

- |                                    |                                  |                                      |                                      |
|------------------------------------|----------------------------------|--------------------------------------|--------------------------------------|
| <input type="checkbox"/> A ./a/a/b | <input type="checkbox"/> E ./b   | <input type="checkbox"/> I ./a/c/b/a | <input type="checkbox"/> M ./b/a/a/c |
| <input type="checkbox"/> B ./a/b/c | <input type="checkbox"/> F ./b/b | <input type="checkbox"/> J ./b/a/c   | <input type="checkbox"/> N ./c       |
| <input type="checkbox"/> C ./c/a   | <input type="checkbox"/> G ./a/c | <input type="checkbox"/> K ./a       | <input type="checkbox"/> O ./a/a     |
| <input type="checkbox"/> D ./a/a/c | <input type="checkbox"/> H ./a/b | <input type="checkbox"/> L ./b/a     | <input type="checkbox"/> P ./a/a/d   |

**Question 11 ♣** Même question pour « `find . -type f -exec cat {} \;` »

- |                                  |   |                                |                                   |
|----------------------------------|---|--------------------------------|-----------------------------------|
| <input type="checkbox"/> A ./a/c | <input type="checkbox"/> C No such file | <input type="checkbox"/> E c   | <input type="checkbox"/> G cat \; |
| <input type="checkbox"/> B error | <input type="checkbox"/> D ./c          | <input type="checkbox"/> F cat | <input type="checkbox"/> H b      |

## 5 Permissions

**Question 12 ♣** L'utilisateur `tata` entre la commande « `rm /home/toto/fichier` » mais cela affiche le message d'erreur « `rm: impossible de supprimer /home/toto/fichier: Permission non accordée` ». Parmi les raisons suivantes, indiquez toutes celles qui peuvent être la cause du message d'erreur.

- |   |   |
|---|---|
| <input type="checkbox"/> A <code>tata</code> ne peut pas exécuter (x) <code>/home/toto/fichier</code> | <input type="checkbox"/> G Le fichier <code>/home/toto/fichier</code> n'existe pas        |
| <input type="checkbox"/> B <code>tata</code> ne peut pas traverser (x) <code>/home/toto</code>        | <input type="checkbox"/> H Le répertoire <code>/home/tata</code> n'existe pas             |
| <input type="checkbox"/> C <code>tata</code> ne peut pas lire (r) <code>/home/toto/fichier</code>     | <input type="checkbox"/> I Le répertoire <code>/home/toto</code> n'existe pas             |
| <input type="checkbox"/> D <code>tata</code> ne peut pas modifier (w) <code>/home</code>              | <input type="checkbox"/> J <code>tata</code> ne peut pas lire (r) <code>/home/toto</code> |
| <input type="checkbox"/> E <code>tata</code> ne peut pas modifier (w) <code>/home/toto</code>         | <input type="checkbox"/> K <code>tata</code> ne peut pas lire (r) <code>/home</code>      |
| <input type="checkbox"/> F <code>tata</code> ne peut pas modifier (w) <code>/home/toto/fichier</code> | <input type="checkbox"/> L <code>tata</code> ne peut pas traverser (x) <code>/home</code> |

**Question 13 ♣** Le fichier `toto` a les permissions `rwrx-x---`. Parmi les commandes suivantes, lesquelles passent les droits de `toto` à `rw-r-----` ?

- |  |   |   |
|--|---|---|
| <input type="checkbox"/> A <code>chmod og+r toto</code>      | <input type="checkbox"/> E <code>chmod a+r toto</code>  | <input type="checkbox"/> I <code>chmod a-x toto</code>  |
| <input type="checkbox"/> B <code>chmod 421 toto</code>       | <input type="checkbox"/> F <code>chmod ug-x toto</code> | <input type="checkbox"/> J <code>chmod a=r toto</code>  |
| <input type="checkbox"/> C <code>chmod 649 toto</code>       | <input type="checkbox"/> G <code>chmod +1 toto</code>   | <input type="checkbox"/> K <code>chmod rwr toto</code>  |
| <input type="checkbox"/> D <code>chmod rw-r----- toto</code> | <input type="checkbox"/> H <code>chmod 640 toto</code>  | <input type="checkbox"/> L <code>chmod go=r toto</code> |

## 6 Liens symboliques

À partir du répertoire `/dir`, on trouve 9 entrées : 5 sous-répertoires et 4 liens symboliques.

<code>/dir:</code>	<code>/dir/tata:</code>	<code>/dir/toto:</code>
<code>tata/</code>	<code>tata -&gt; ..</code>	<code>tata/</code>
<code>toto/</code>	<code>toto/</code>	<code>toto/</code>
<code>tutu -&gt; tata</code>	<code>tutu -&gt; ../toto</code>	<code>tutu -&gt; ../tata/tutu/../../tutu</code>

**Question 14** Que se passe-t-il si l'on fait `cd tutu/toto` à partir du répertoire `/dir` ?

- |  |   |
|--|---|
| <input type="checkbox"/> A On se retrouve physiquement dans <code>/dir/tata/tata</code>    | <input type="checkbox"/> G Cela affiche l'erreur « Aucun fichier de ce type »           |
| <input type="checkbox"/> B On se retrouve physiquement dans <code>/dir/toto</code>         | <input type="checkbox"/> H On se retrouve physiquement dans <code>/</code>              |
| <input type="checkbox"/> C Cela affiche l'erreur « Ce n'est pas un dossier »               | <input type="checkbox"/> I On se retrouve physiquement dans <code>/dir/toto/tata</code> |
| <input type="checkbox"/> D On se retrouve physiquement dans <code>/dir/tata</code>         | <input type="checkbox"/> J On se retrouve physiquement dans <code>/dir/toto/toto</code> |
| <input type="checkbox"/> E On reste physiquement et silencieusement dans <code>/dir</code> | <input type="checkbox"/> K Cela affiche l'erreur « Permission non accordée »            |
| <input type="checkbox"/> F Cela affiche l'erreur « Trop de liens symboliques »             | <input type="checkbox"/> L On se retrouve physiquement dans <code>/dir/tata/toto</code> |

**Question 15** Que se passe-t-il si l'on fait `cd tutu/tutu/tutu` à partir du répertoire `/dir` ?

- |   |  |
|---|--|
| <input type="checkbox"/> A On se retrouve physiquement dans <code>/dir/toto</code>      | <input type="checkbox"/> G On se retrouve physiquement dans <code>/</code>                 |
| <input type="checkbox"/> B On se retrouve physiquement dans <code>/dir/toto/tata</code> | <input type="checkbox"/> H Cela affiche l'erreur « Permission non accordée »               |
| <input type="checkbox"/> C Cela affiche l'erreur « Aucun fichier de ce type »           | <input type="checkbox"/> I On reste physiquement et silencieusement dans <code>/dir</code> |
| <input type="checkbox"/> D On se retrouve physiquement dans <code>/dir/toto/toto</code> | <input type="checkbox"/> J On se retrouve physiquement dans <code>/dir/tata/tata</code>    |
| <input type="checkbox"/> E Cela affiche l'erreur « Ce n'est pas un dossier »            | <input type="checkbox"/> K On se retrouve physiquement dans <code>/dir/tata</code>         |
| <input type="checkbox"/> F Cela affiche l'erreur « Trop de liens symboliques »          | <input type="checkbox"/> L On se retrouve physiquement dans <code>/dir/tata/toto</code>    |

**Question 16** À partir du répertoire `dir`, quelle commande crée un lien symbolique `/dir/toto/toto/titi` qui mène au répertoire `/dir/tata/toto` ?

- |   |   |
|---|---|
| <input type="checkbox"/> A <code>ln -s tata/toto toto/toto/titi</code>          | <input type="checkbox"/> H <code>ln -s toto/toto/titi ../../tata</code>         |
| <input type="checkbox"/> B <code>ln -s toto/toto/titi ../../tata/toto</code>    | <input type="checkbox"/> I <code>ln -s toto/toto/titi ../tata</code>            |
| <input type="checkbox"/> C <code>ln -s toto/toto/titi tata/toto</code>          | <input type="checkbox"/> J <code>ln -s tata toto/toto/titi</code>               |
| <input type="checkbox"/> D <code>ln -s toto/toto/titi ../../../tata/toto</code> | <input type="checkbox"/> K <code>ln -s toto/toto/titi ../tata/toto</code>       |
| <input type="checkbox"/> E <code>ln -s ../../tata/toto toto/toto/titi</code>    | <input type="checkbox"/> L <code>ln -s ../../tata toto/toto/titi</code>         |
| <input type="checkbox"/> F <code>ln -s ../tata toto/toto/titi</code>            | <input type="checkbox"/> M <code>ln -s ../../../tata/toto toto/toto/titi</code> |
| <input type="checkbox"/> G <code>ln -s toto/toto/titi tata</code>               | <input type="checkbox"/> N <code>ln -s ../tata/toto toto/toto/titi</code>       |

## 7 Chercher dans des fichiers

Le fichier `poeme` contient un extrait de poème.

```
$ cat poeme
Il est terrible
le petit bruit de l'oeuf dur cassé sur un comptoir d'étain
il est terrible ce bruit
quand il remue dans la mémoire de l'homme qui a faim
```

**Question 17** ♣ Quel est le résultat de la commande « `grep -c il poeme` » ?

A 2  
 B 1

C poeme  
 D 3

E il  
 F 1

G i  
 H 4

**Question 18** ♣ Lesquelles des commandes suivantes permettent d'afficher uniquement la première ligne du poème ?

A `grep 'Il est' poeme`  
 B `tail -n 1`  
 C `grep terrible poeme`  
 D `grep -v u poeme`

E `head -n 1 poeme`  
 F `grep -v terrible poeme`  
 G `grep -B 1 oeuf poeme | head -n 1`  
 H `grep -B 1 oeuf poeme`

## 8 Expressions régulières

**Question 19** ♣ Quelles expressions régulières POSIX basique (BRE) trouvent une **correspondance** avec le mot « violon » mais pas avec le mot « avion » ?

A `[a-m][a-m]`  
 B `o.o`  
 C `1`

D `o*o`  
 E `[^avion]`  
 F `on$`

G `[violon]`  
 H `^[^a]*$`  
 I `[v][i][o][l][o][n]`

**Question 20** ♣ Parmi les mots suivants, lesquels ont une **correspondance** avec l'expression régulière POSIX basique (BRE) « `emm*[a-f].[^ta]$` » ?

A remembré  
 B emmuré  
 C emmené

D dilemmes  
 E jumeaux  
 F écrémage

G emmena  
 H emmailloter  
 I flemmard



## Annexe détachable

**cat** Concaténer des fichiers et les afficher sur la sortie standard.

**chmod** Modifier les bits de comportement de fichier.

`chmod [OPTION]... MODE[,MODE]... FICHIER...`

`chmod [OPTION]... MODE-OCTAL FICHIER`

Le format d'un mode symbolique est `[ugoa...][[-+=][rwx...].` Plusieurs modes symboliques peuvent être indiqués ensemble, séparés par des virgules.

Une combinaison des lettres `ugoa` contrôle la catégorie d'accès à modifier. Il peut s'agir de l'utilisateur possédant le fichier (`u`), des autres utilisateurs du même groupe que le fichier (`g`), des utilisateurs n'appartenant pas au groupe du fichier (`o`), ou de tous les utilisateurs (`a`). Si aucune catégorie n'est indiquée, `a` est pris par défaut, mais les bits correspondant à des bits à 1 de `l`umask ne sont pas affectés.

L'opérateur `+` ajoute à chaque fichier les bits de comportement de fichier spécifiés à ceux déjà existants, l'opérateur `-` les enlève, et l'opérateur `=` les ajoute tout en supprimant ceux qui n'ont pas été indiqués. Dans ce dernier cas, les bits d'emprunt des identifiants d'utilisateur et de groupe d'un répertoire ne sont toutefois pas affectés.

Les lettres `rwx` sélectionnent les bits de comportement de fichier des utilisateurs concernés : lecture (`r`), écriture (`w`), exécution (ou recherche pour les répertoires) (`x`).

La valeur pour le mode numérique est constituée de 1 à 4 chiffres en octal (0 à 7), calculés en additionnant les bits de valeurs 4, 2 et 1, tout bit non mentionné étant considéré comme des zéros en début de chaîne. Le premier chiffre correspond aux attributs identifiant de l'utilisateur (4), identifiant du groupe (2) et suppression restreinte ou bit sticky (1). Le second chiffre correspond aux autorisations de lecture (4), écriture (2) et exécution (1) pour l'utilisateur propriétaire du fichier. Les troisième et quatrième chiffres correspondent respectivement aux autorisations d'accès pour les membres du groupe du fichier, et pour les autres utilisateurs. Les valeurs ont les mêmes significations que pour le deuxième chiffre.

**cut** Supprimer une partie de chaque ligne d'un fichier.

`-c LISTE` sélectionner seulement ces caractères.

`-f LISTE` sélectionner seulement ces champs.

`LISTE` se compose d'un intervalle ou de plusieurs intervalles séparés par des virgules. Chaque intervalle se compose de : « `N` » Nième octet, caractère ou champ, compté à partir de 1 ; « `N-` » du Nième octet, caractère ou champ jusqu'à la fin de la ligne ; « `N-M` » du Nième au Mième octet (inclus), caractère ou champ ; « `-M` » du premier au Mième octet (inclus), caractère ou champ.

**echo** Afficher une ligne de texte.

**find** Rechercher des fichiers dans une hiérarchie de répertoires

`-type C` fichier du type `C` : « `b` » fichier spécial en mode bloc ; « `c` » fichier spécial en mode caractère ; « `d` » répertoire ; « `f` » fichier régulier ; « `l` » lien symbolique.

`-exec commande` ; Exécuter la *commande*. Tous les paramètres qui suivent `find` sont considérés comme des paramètres pour la ligne de commande, jusqu'à la rencontre d'un caractère « ; ». La chaîne « `{}` » est remplacée par le nom du fichier en cours de traitement. Ces deux chaînes peuvent

avoir besoin d'être protégées du développement de la ligne de commande par le shell, en utilisant le caractère d'échappement (« `\` ») ou une protection par des guillemets. La commande indiquée est exécutée à chaque fois qu'un fichier correspond. La commande est exécutée depuis le répertoire de départ.

**grep** Rechercher les lignes correspondant à un motif donné. Par défaut, `grep` affiche les lignes qui contiennent une correspondance au motif.

`-v, --invert-match` Inverser la mise en correspondance, pour sélectionner les lignes ne correspondant pas au motif.

`-c, --count` Ne pas afficher les résultats normaux. À la place, afficher un décompte des lignes correspondant au motif pour chaque fichier.

`-B N, --before-context=N` Afficher les `N` lignes qui précèdent celle qui contient le motif.

**head** Afficher le début des fichiers.

`-n, --lines=K` afficher les `K` premières lignes au lieu des 10 premières.

**ln** Créer des liens entre des fichiers.

`ln [OPTION]... CIBLE NOM` (1er format)

`ln [OPTION]... CIBLE` (2e format)

`ln [OPTION]... CIBLE... RÉPERTOIRE` (3e format)

`ln [OPTION]... -t RÉPERTOIRE CIBLE...` (4e format)

Dans le 1er format, créer un lien vers la CIBLE avec le NOM. Dans le 2e format, créer un lien vers la CIBLE dans le répertoire actuel. Dans les 3e et 4e formats, créer des liens vers chaque CIBLE dans le RÉPERTOIRE. Par défaut, les liens créés sont des liens durs ; pour créer des liens symboliques, utiliser `-s`. Les noms des liens à créer ne doivent pas, par défaut, déjà exister. Lors de la création de liens durs, chaque CIBLE doit exister. Un lien symbolique peut contenir n'importe quel texte ; lors de la résolution du lien par la suite, un lien relatif est interprété par rapport à son répertoire parent.

**ls** Afficher le contenu de répertoires.

`-l` utiliser un format d'affichage long.

`-i` afficher le numéro d'inœud de chaque fichier.

**mkdir** Créer des répertoires.

`-p, --parents` créer des répertoires parent si nécessaire, sans générer d'erreur s'ils existent.

**rm** Effacer des fichiers et des répertoires

`-r, -R, --recursive` enlever le contenu des répertoires récursivement. Par défaut, `rm` n'efface pas les répertoires. Utilisez cette option pour effacer chaque répertoire passé en paramètre en même temps que son contenu.

**sort** Trier les lignes de fichiers texte.

`-n, --numeric-sort` comparer selon la valeur numérique de la chaîne

**tail** Afficher la dernière partie de fichiers.

`-n, --lines=K` afficher les `K` dernières lignes, au lieu des 10 dernières.

**uniq** Éliminer les lignes successives identiques.

`-c, --count` préfixer les lignes par le nombre d'occurrences.



0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1
2	2	2	2	2	2	2	2
3	3	3	3	3	3	3	3
4	4	4	4	4	4	4	4
5	5	5	5	5	5	5	5
6	6	6	6	6	6	6	6
7	7	7	7	7	7	7	7
8	8	8	8	8	8	8	8
9	9	9	9	9	9	9	9

← Codez les 8 chiffres de votre code permanent ci-contre, et inscrivez-le à nouveau ci-dessous avec vos nom et prénom.

Code permanent :
.....
Prénom :
.....
Nom :
.....

*Les réponses aux questions sont à donner exclusivement sur cette feuille : les réponses données sur les feuilles précédentes ne seront pas prises en compte.*

*Important : noircissez complètement l'intérieur de chaque case (pas de croix, pas de cercles).*

- Question 1 : 

A	B	C	D	E	F	G	H	I
---	---	---	---	---	---	---	---	---
- Question 2 : 

A	B	C	D	E	F	G	H	I	J
---	---	---	---	---	---	---	---	---	---
- Question 3 : 

A	B	C	D	E
---	---	---	---	---
- Question 4 : 

A	B	C	D	E	F	G	H	I	J	K	L
---	---	---	---	---	---	---	---	---	---	---	---
- Question 5 : 

A	B	C	D	E	F	G	H	I	J	K	L
---	---	---	---	---	---	---	---	---	---	---	---
- Question 6 : 

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---
- Question 7 : 

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---
- Question 8 : 

A	B	C	D	E	F	G	H	I	J	K	L
---	---	---	---	---	---	---	---	---	---	---	---
- Question 9 : 

A	B	C	D	E	F	G	H	I	J	K	L
---	---	---	---	---	---	---	---	---	---	---	---
- Question 10 : 

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---
- Question 11 : 

A	B	C	D	E	F	G	H
---	---	---	---	---	---	---	---
- Question 12 : 

A	B	C	D	E	F	G	H	I	J	K	L
---	---	---	---	---	---	---	---	---	---	---	---
- Question 13 : 

A	B	C	D	E	F	G	H	I	J	K	L
---	---	---	---	---	---	---	---	---	---	---	---
- Question 14 : 

A	B	C	D	E	F	G	H	I	J	K	L
---	---	---	---	---	---	---	---	---	---	---	---
- Question 15 : 

A	B	C	D	E	F	G	H	I	J	K	L
---	---	---	---	---	---	---	---	---	---	---	---
- Question 16 : 

A	B	C	D	E	F	G	H	I	J	K	L	M	N
---	---	---	---	---	---	---	---	---	---	---	---	---	---
- Question 17 : 

A	B	C	D	E	F	G	H
---	---	---	---	---	---	---	---
- Question 18 : 

A	B	C	D	E	F	G	H
---	---	---	---	---	---	---	---
- Question 19 : 

A	B	C	D	E	F	G	H	I
---	---	---	---	---	---	---	---	---
- Question 20 : 

A	B	C	D	E	F	G	H	I
---	---	---	---	---	---	---	---	---