

Licence 2 - Informatique
INF3B1 - Système Linux
Durée 1 heure 30 mn

Chaque candidat doit, au début de l'épreuve, porter son nom dans le coin de la copie qu'il cachera par collage après avoir été pointé. Il devra en outre porter son numéro de place sur chacune des copies, intercalaires, ou pièces annexées.

Ce sujet est constitué du présent document, agrafé. Chacune des quatre parties est à rendre sur une copie séparée.

Tout document autorisé.

1 Expressions régulières

Cette partie qui comprend les questions 1 et 2 est à rendre sur une copie séparée

Question 1.

On exécute des commandes du type `cat /usr/share/dict/words | grep -E "motif"`.
Pour chaque commande ci-dessous, préciser quelle est la forme des chaînes de caractères décrites par l'expression régulière `motif` en explicitant les notations et signes utilisés par cette expression.
Attention à la rédaction, seules les réponses claires et non ambiguës seront prises en compte.

1.a)

```
$ cat /usr/share/dict/words | grep -E "kp[^a-o]"
shockproof
```

1.b)

```
$ cat /usr/share/dict/words | grep -E "nobbi?er"
knobbier
snobbery
snobbery's
snobbier
```

1.c)

```
$ cat /usr/share/dict/words | grep -E "ckp..$"
chickpea
cockpit
crackpot
jackpot
stickpin
```

1.d)

```
$ cat /usr/share/dict/words | grep -E "^j[~io]*n$"
japan
jellybean
```

1.e)

```
$ cat /usr/share/dict/words | grep -E "(ou).*\1.*o"
Souphanouvong
Souphanouvong's
Yamoussoukro
```

Question 2.

Écrire les commandes qui permettent de sélectionner dans le fichier `/usr/share/dict/words` :

- 2.a) les mots qui finissent par `system` ;
- 2.b) les mots qui ne contiennent pas la lettre `a` ;
- 2.c) les mots dont le troisième caractère est un `c` ;
- 2.d) les mots qui ne contiennent que des majuscules ;
- 2.e) les mots qui contiennent deux `w` ou deux `z` .

2 L'utilitaire AWK

Cette partie qui comprend les questions 3 et 4 est à rendre sur une copie séparée

Question 3.

On dispose d'un fichier notes.txt contenant les notes de chaque étudiant. On se limite à deux notes par étudiant.

```
$cat notes.txt
Bob 10.5 12
Alice 15 18
Marie 10 9
Eric 14 18 ...
```

Afficher la note moyenne de chaque étudiant en utilisant l'utilitaire AWK.

```
$awk "à définir" notes.txt
Bob:11.25
Alice:16.5
Marie:9.5
Eric:16
...
```

Compléter la question précédente pour afficher le nombre des étudiants ayant une note moyenne supérieure à 10 par rapport au nombre total des étudiants.

```
$awk "déjà définis précédemment" notes.txt | awk "à définir"
Il y a 3/4 des étudiants ayant une note moyenne plus que 10
```

Question 4.

Écrire un script .awk permettant d'afficher le nombre de mots respectant le motif passé en argument du script dans chaque phrase du fichier mesPhrases.txt.

Indication : `$mot ~ motif` retourne 1 si le **mot** respecte le **motif** (expression régulière).

```
$ cat mesPhrases.txt
Bonjour tout le monde
ou pourrais-je partir ?
Au revoir !
$ awk -f script.awk mesPhrases.txt "ou."
Phrase 1 : 2 mots
Phrase 2 : 1 mots
Phrase 3 : 0 mots
$ cat script.awk
BEGIN { "À définir : récupérer le motif, etc." }
{
    for(i = 1; i <= "à définir : nombre de mots dans la phrase courante"; i++)
    { "À définir" }
    "À définir : afficher le nombre de mots respectant le motif dans la phrase courante"
}
```

3 Bash

Cette partie qui comprend les questions 5 à 7 est à rendre sur une copie séparée

Le code de la figure 1 présente deux façons différentes pour produire la chaîne de caractères de la variable `exclure` à partir de la variable `extensions`.

```
0  #!/bin/bash
1
   #----- solution 1 -----
   extensions="*.aux *.bbl *.blg"
   exclure=""
5  for i in $extensions ; do
       exclure="$exclure --exclude=$i"
   done
   echo $exclure

10 #----- solution 2 -----
   extensions="aux bbl blg"
   exclure=""
   for i in $extensions ; do
       exclure="$exclure --exclude=*. $i"
15 done
   echo $exclure
```

Figure 1: Script `sujet.sh`

Son exécution fournit le résultat suivant :

```
$ ls
sujet.aux  sujet.log  sujet.pdf  sujet.sh  sujet.tex
$ bash sujet.sh
--exclude=sujet.aux --exclude=*.bbl --exclude=*.blg
--exclude=*.aux --exclude=*.bbl --exclude=*.blg
```

Question 5. Expliquez comment la variable `exclure` est produite.

Question 6. Quelle est la bonne solution ? Quelles sont ses limitations ? Expliquez pourquoi la mauvaise solution ne fonctionne pas.

Question 7. En utilisant `sed`, proposez une modification du code qui remplace la boucle `for`.

4 Script Shell

Cette partie qui comprend les questions 8 à 10 est à rendre sur une copie séparée

Créer un script Shell qui prend en paramètre un répertoire quelconque **rep** et qui permet de :

Question 8.

Créer trois répertoires : rep_fichiers_non_executables, sous_rep et rep_fichiers_executables.

Question 9.

- 9.a) Calculer le nombre de fichiers non exécutables, de sous répertoires et de fichiers exécutables du répertoire **rep** ;
- 9.b) Déplacer chaque élément de chaque type (fichier non exécutable, sous répertoire ou fichier exécutable) du répertoire **rep** dans le répertoire approprié (créé en question 8).

Indication : l'option **-x** de la commande test sert à identifier un fichier exécutable.

Question 10.

Supprimer le répertoire initial **rep** s'il est désormais vide.

Exemple :

```
$ ls -l monDossier/rep/ | awk 'print $1 " " " $NF'
-rwxr--r--  exercice.awk
-rw-r--r--  file1.txt
-rw-r--r--  file2.pdf
-rwxr--r--  script.sh
drwxr-xr-x  sousrep1
drwxr-xr-x  sousrep2
drwxr-xr-x  sousrep3

$ ScriptExamen.sh monDossier/rep
Il y a 3 répertoires, 2 fichiers non exécutables et 2 fichiers exécutables dans rep.

$ ls -R
./rep_fichiers_non_executables:
file1.txt file2.pdf

./rep_fichiers_executables:
exercice.awk script.sh

./sous_rep:
sousrep1 sousrep2 sousrep3

./monDossier:
```