# TD 4 : Expressions régulières et programmation AWK

## Exercice 1 : Expression régulière

1. Soit le tableau de chaînes suivant :

| Numéro | Chaîne |
|---|---|
| a | abc |
| b | zzzz xx |
| c | abcdef |
| d | 123456 7890 abcaziuz |
| e | yyyy |
| f | xyz stuv abc |
| g | xx abcxxxxxxxxx |
| h | xAb* 12345 |
| i | xAB* 45678 |
| j | 98745 xaB* 23654 |
| k | abc$!k; |
| l | 567 |
| m | 5666777 |
| n | 57 |
| o | Suite... du paragraphe |
| p | Suite... de l'histoire |
| q | la suite... |
| r | Suite.. au prochain numero. |

Table 1: **Chaînes de caractères**

Pour chaque expression régulières ci dessous, trouver la ou les chaînes du tableau ci-dessus satisfaisant à l'expression régulière :

| | Expression Régulière |
|---|---|
| 1 | c$ |
| 2 | c\$ |
| 3 | ^abc |
| 4 | abc$ |
| 5 | ^abc$ |
| 6 | ^abc. |
| 7 | ^56[67] |
| 8 | .56[67] |
| 9 | x[Aa][Bb] |
| 10 | x[^Aa] |
| 11 | [Aa][^b] |
| 12 | abcd* |
| 13 | 566?7 |

| | Expression Régulière |
|---|---|
| 14 | [r-v] |
| 15 | 56*7* |
| 16 | 56+7+ |
| 17 | 56?7? |
| 18 | 566?7 |
| 19 | 987\|789 |
| 20 | abc\|[def] |
| 21 | .*[Aa][Bb].*12.* |
| 22 | .*12.*[Aa][Bb] |
| 23 | .*[Aa]b.*12.*\|.*12.*[Aa]b.* |
| 24 | .*([Aa]b.*12.*\|.*12.*[Aa]b).* |
| 25 | abc[def][m-x]* |
| 26 | ^[Ss]uite\.\.\..* |

Table 2: **Expressions régulières**

Réponse : Question 1

```
1 - a                   14 - o, p, r
2 - k                   15 - d, j, l, m, n
3 - a, c, k             16 - l, m
4 - a                   17 - d, j, l, m, n
5 - a                   18 - l
6 - c, k                19 - j
7 - l, m                20 - a, c, k, o, p, r
8 - i, l, m             21 - h
9 - h, i                22 - d
10 - f, g               23 - d, h
11 - i                  24 - d, h
12 - a, c, k            25 - c
13 - l                  26 - o, p
```

2. Construire une expression régulière qui contient au moins "ab" suivi d'un nombre quelconque du caractère "b".

   **réponse** : `abb*`

3. Ecrire la commande **`egrep`** pour chacune des tâches suivantes :

   (a) Trouver toutes les chaînes qui contiennent "**`ted`**" ou "**`fred`**"

   (b) Trouver toutes les chaînes qui contiennent "**`ed`**", "**`ted`**" ou "**`fed`**"

   (c) Trouver toutes les chaines qui ne commencent pas par "**`g`**"

   (d) Trouver toutes les chaines commençant par "**`g`**" ou n'importe quel chiffre de **`0`** à **`9`**

   (e) Trouver toutes les chaines commençant par "**`guna`**"

   (f) Trouver toutes les lignes d'un fichier où la chaine de caractère "**`sam`**" est répétée au moins deux fois

   (g) Trouver toutes les lignes d'un fichier contenant une adresse mail

Réponse : Question 3

```
a) \(t\|fr\)ed
b) [tf]?ed
c) ^[^g]
d) ^[g0-9]
e) ^guna
f) sam.*sam
g) ^[a-zA-Z0-9._]+@[a-zA-Z0-9.-]+\.[a-zA-Z]\{2,4\}$
```

4. Ecrire une expression régulière reconnaissant les nombres compris entre **`1000`** et **`9999`**

Réponse : Question 4

```
^[1-9][0-9]\{3\}$
```

5. Ecrire une expression régulière reconnaissant les nombres compris entre **`100`** and **`9999`**

Réponse : Question 5

```
^[0-9][0-9]\{2,3\}$
```

6. Ecrire une expression régulière qui permet de reconnaitre tous les fichiers d'extension **`.txt`** du répertoire courant qui ont été modifiés en Novembre

Réponse : Question 6

```
ls -l | egrep Nov\ .*\.txt$
```

## Exercice 2 : Programmation AWK

1 - Soit le fichier **notes** ayant le contenu suivant :

```
BOB:18:8:15
MARIE:6:11:0
PIERRE:15:0:20
LUCIE:16:16:17
LOUISE:10:20:15
PAUL:16:18:12
```

(a) Donner le résultat pour chacune des commandes suivantes :

```
$ awk -F: '{print $0}' notes
```

Réponse

```
BOB:18:8:15
MARIE:6:11:0
PIERRE:15:0:20
LUCIE:16:16:17
LOUISE:10:20:15
PAUL:16:18:12
```

```
$ awk -F: '{print $1}' notes
```

Réponse

```
BOB
MARIE
PIERRE
LUCIE
LOUISE
PAUL
```

```
$ awk -F: '{print $4}' notes
```

Réponse

```
15
0
20
17
15
12
```

(b) Calculer la moyenne des notes par étudiant (afficher **Nom : moyenne**) et donner la moyenne générale de toute la classe (**Moyenne générale : moy**).

Réponse

```
awk -F: 'BEGIN {print "Nom : Moyenne"}
               {moyenne=($2+$3+$4)/3
               somme+= moyenne
               print $1 " : " moyenne}
        END{print "Moyenne générale : ", somme/NR}' notes
```

(c) Ecrire un script Shell (**ExoAWK**) qui va afficher les notes d'un étudiant, dont son nom est donné en paramètre. (Exemple de résultat d'exécution : **./ExoAWK MARIE** qui donne comme résultat :

```
Recherche de : MARIE
2 : MARIE   6   11   0
```

Réponse

```
#!/bin/bash
if [ $# -eq 1 ]
then
    awk -F: -v nom=$1 'BEGIN {print "Recherche de : " nom}
                $1~/^nom$/{print NR " : " $1,$2,$3,$4}' notes
fi
```

II - Soit le fichier **pstxt** ayant le contenu suivant : Affichez le **UID**, le **PID** et la commande **COMMAND** d'un utilisateur

Table 3: Fichier **pstxt**

| UID | PID | PPID | TIME | COMMAND |
|---|---|---|---|---|
| 0 | 1 | 0 | 0:06.95 | /sbin/launchd |
| 0 | 75 | 1 | 0:00.19 | BBDaemon |
| 0 | 76 | 1 | 0:00.03 | /usr/libexec/wdhelper |
| 0 | 87 | 1 | 0:02.53 | /usr/sbin/securityd -i |
| 205 | 89 | 1 | 0:01.00 | /usr/libexec/locationd |
| 0 | 92 | 1 | 0:00.13 | /usr/sbin/blued |
| 0 | 93 | 1 | 0:00.02 | autofsd |
| 65 | 97 | 1 | 0:00.57 | /usr/sbin/mDNSResponder |
| 241 | 108 | 1 | 0:00.68 | /usr/sbin/distnoted daemon |
| 202 | 163 | 1 | 0:00.57 | /usr/sbin/coreaudiod |
| 24 | 208 | 1 | 0:00.62 | /usr/libexec/networkd |
| 70 | 222 | 84 | 0:00.00 | /usr/sbin/httpd -D FOREGROUND |
| 242 | 243 | 1 | 0:00.04 | /usr/libexec/nsurlsessiond --privileged |
| 24 | 252 | 1 | 0:00.98 | /usr/libexec/symptomsd |
| 501 | 414 | 1 | 0:00.85 | /usr/libexec/secd |

passé en paramètre (la commande **id -u User** donne l'**UID** associé à l'utilisateur **User**).

Réponse

```
#!/bin/bash

if [ $# -eq 1 ]; then
    # UID=`id -u $1` récupérer l'UID de l'utilisateur passé en paramètre $1
    awk -v UID=`id -u $1` '$1==UID {print $1, $2, $5}' pstxt
else
    echo "Nombre de paramètre incorrecte"
    exit 1
fi
```