ED 3 – Expressions régulières correction

EXERCICE 1

Soit le tableau de chaînes suivant :

commande1	grep c$ fichexpregulbis	abc	a
		xyzstuvabc	f
commande2	grep 'c\$' fichexpregulbis	Abc$!k ;	k
commande3	grep ^abc fichexpregulbis	abc	a
		abcdef	c
		abc$!k;	k
commande4	grep abc$ fichexpregulbis	abc	a
		xyzstuvabc	f
commande5	grep ^abc$ fichexpregulbis	abc	a
commande6	grep ^abc. fichexpregulbis	abcdef	c
		abc$!k;	k
commande7	grep 45 fichexpregulbis	1234567890abcaziuz	d
		xAb*12345	h
		xAB*45678	i
		98745xaB*23654	j
commande8	grep ^56[67] fichexpregulbis	567	l
		5666777	m
commande9	grep .56[67] fichexpregulbis	1234567890abcaziuz	d
		xAB*45678	i
commande10	grep x[Aa][Bb] fichexpregulbis	xxabcxxxxxxxxx	g
		xAb*12345	h
		xAB*45678	i
		98745xaB*23654	j
commande11	grep x[^Aa] fichexpregulbis	zzzzxx	b
		xyzstuvabc	f
		xxabcxxxxxxxxx	g
commande12	grep [Aa][^b] fichexpregulbis	1234567890abcaziuz	d
		xAB*45678	i
		98745xaB*23654	j

		Suite... du paragraphe	o
		la suite...	q
		Suite.. au prochain	r
commande13	grep abcd* fichexpregulbis	abc	a
		abcdef	c
		1234567890abcaziuz	d
		xyzstuvabc	f
		xxabcxxxxxxxxx	g
		abc$!k;	k
commande14	grep -E 566?7 fichexpregulbis	1234567890abcaziuz	d
		567	l
		xAB*45678	i
commande15	grep [r-v] fichexpregulbis	1234567890abcaziuz	d
		xyzstuvabc	f
		Suite... du paragraphe	o
		Suite... de l'histoire	p
		la suite...	q
		Suite.. au prochain numero.	r
commande16	grep -E 56*7* fichexpregulbis	1234567890abcaziuz	d
		xAb*12345	h
		xAB*45678	i
		98745xaB*23654	j
		567	l
		5666777	m
		57	n
commande17	grep -E 56+7+ fichexpregulbis	1234567890abcaziuz	d
		xAB*45678	i
		567	l
		5666777	m
commande18	grep -E 56?7? fichexpregulbis	1234567890abcaziuz	d
		xAb*12345	h
		xAB*45678	i
		98745xaB*23654	j
		567	l
		57	n
commande19	grep -E 566?7 fichexpregulbis	1234567890abcaziuz	d
		xAB*45678	i
		567	l

commande20	grep -E '987|789' fichexpregulbis	1234567890abcaziuz	d
		98745xaB*23654	j
commande21	grep -E 'abc|[def]' fichexpregulbis	abc	a
		abcdef	c
		1234567890abcaziuz	d
		xyzstuvabc	f
		xxabcxxxxxxxxx	g
		abc$!k;	k
		Suite... du paragraphe	o
		Suite... de l'histoire	p
		la suite...	q
		Suite.. au prochain numero.	r
commande22	grep -E '.*[Aa][Bb].*12.*' fichexpregulbis	xAb*12345	h
commande23	grep -E '.*12.*[Aa][Bb]' fichexpregulbis	1234567890abcaziuz	d
commande24	grep -E '.*[Aa]b.*12.*|.*12.*[Aa]b.*' fichexpregulbis	1234567890abcaziuz	d
		xAb*12345	h
commande25	grep -E '.*([Aa]b.*12.*|.*12.*[Aa]b).*' fichexpregulbis	1234567890abcaziuz	d
		xAb*12345	h
commande26	grep -E 'abc[def][m-x]*' fichexpregulbis	abcdef	c

EXERCICE 2

1. **Chercher toutes les lignes commençant par «a» ou «A»**.

Il faut indiquer que l'on veut le début de la ligne, avec le chapeau (*caret* en anglais). Ensuite, on veut préciser que la ligne commence par un «a» minuscule ou majuscule. Il y a deux façons de le faire :

- Utiliser l'option `-i` qui fait ignorer la différence entre les majuscules et le minuscules.
- Dire que l'on cherche un «a» ou un «A». C'est à cela que servent les crochets : `[abc]` signifie «a ou b ou c». Ici, ce sera `[aA]`.

Enfin, il faut protéger les signes contre le shell, pour qu'il ne les interprète pas; on met donc l'expression entre apostrophes. Il faut donc écrire :

```
grep -i '^a' fichier
```

ou

```
grep '^[aA]' fichier
```

2. **Chercher toutes les lignes finissant par «rs»**

C'est le dollar (`$`) qui représente la fin de la ligne. Il faut donc écrire :

```
grep 'rs$' fichier
```

3. **Chercher toutes les lignes contenant au moins un chiffre**

Pour désigner un chiffre, on peut en indiquer une liste entre crochets : `[0123456789]`. Il est plus simple d'utiliser une classe de caractères : `[0-9]` qui désigne, comme la solution précédente, n'importe quel chiffre de zéro à neuf.

Il faut donc taper :

```
grep '[0-9]' fichier
```

4. **Chercher toutes les lignes commençant par une majuscule**

Comme on l'a vu, c'est le chapeau qui indique le début de la ligne. Pour indiquer que l'on cherche une majuscule, on peut soit en donner une liste (`[ABCDEFGHIJKLMNOPQRSTUVWXYZ]`), soit utiliser une classe de caractères : `[A-Z]`, la seconde solution étant, de loin, préférable...

Il faut donc taper :

```
grep '^[A-Z]' fichier
```

5. **Chercher toutes les lignes commençant par «B», «E» ou «Q»**

Il faut indiquer entre crochets les trois lettres recherchées :

```
grep '^[BEQ]' fichier
```

6. **Chercher toutes les lignes finissant par un point d'exclamation**

Le point d'exclamation n'a pas de signification particulière avec `grep`, on peut donc le mettre tel quel :

```
grep '!$' fichier
```

7. **Chercher toutes les lignes ne finissant pas par un signe de ponctuation (point, virgule, point-virgule, deux-points, point d'interrogation, point d'exclamation)**

Il faut donner une liste de caractères, que l'on ne veut pas voir figurer; la liste sera entre crochets, comme on l'a déjà vu, et c'est le chapeau qui signifiera, dans ce contexte, «sauf». Par exemple, si on cherche tous les «a», sauf ceux suivi de «b», «c» ou «t», on écrit :

```
grep 'a[^bct]'
```

Il y a une seconde difficulté, qui vient de ce que certains caractères sont spéciaux avec `grep`. Vous vous doutez que le chapeau est spécial quand il est placé au début de l'expression, et que le dollar l'est quand il est placé en fin d'expression. Dans notre cas :

- Le point désigne n'importe quel caractère.
- Le point d'interrogation signifie «le caractère qui précède apparaît 0 ou 1 fois». Avec `egrep`, il fonctionne tout seul, avec `grep`, il faut le faire précéder d'un backslash pour qu'il fonctionne; par exemple (avec `grep`), pour chercher «charbon» ou «vagabond», on écrit :
  ```
  grep 'ar\?bo' fichier
  ```
  (chercher la suite de lettre «abo» avec un «r» facultatif entre le «a» et le «b»).

Pour que `grep` interprète littéralement ces caractères, et ne les considère plus comme spéciaux, il faut les faire précéder d'un backslash (`\`). Si par exemple vous cherchez toutes les lignes qui se terminent par un point, il faut taper :

```
grep '\.$' fichier
```

Dans notre cas cependant, ces caractères sont protégés par les crochets. On peut donc écrire :

```
grep '[^.,;:?!]$' fichier
```

On peut aussi utiliser l'option `-v`, qui prend toutes les lignes où ne figure pas une chaîne de caractères donnée; dans ce cas, on tape :

```
grep  -v '[.,;:?!]$' fichier
```

8. **Comment chercher tous les mots contenant un «r» précédé de n'importe quelle lettre majuscule ou minuscule ?**

On cherche une chaîne de caractères qui soit indifféremment au début ou au milieu d'un mot. N'importe quelle lettre, ce sont les classes de caractères `[a-zA-Z]` ou `[:alpha:]`, qui sont équivalentes.

Il y a une petite subtilité avec l'emploi de classes du second type; elles désignent un groupe de caractères, et il faut mettre une seconde paire de crochets pour dire «n'importe quel caractère de cette classe prédéfinie». On tape donc au choix :

```
grep '[a-zA-Z]r' fichier'
```

ou

```
grep '[[:alpha:]]r' fichier'
```

Attention, dans ces listes ne sont pas compris les caractères accentués...

9. **Chercher tous les mots dont la seconde lettre est un «r»**.

C'est le symbole `\<` qui désigne un début de mot. La première lettre du mot est indifférente, la seconde est un «r». On écrit donc :

```
grep '\<.r' fichier
```

Il y a cependant un problème avec les caractères accentués, que `grep` considère comme des blancs. Dans ce cas, il vaut mieux procéder autrement : un mot est précédé d'un début de ligne, ou d'un blanc ou d'une

tabulation. Un début de ligne, c'est le chapeau, un blanc ou une tabulation, c'est la classe de caractères `[:space:]`.

On va se servir du pipe (`|`) qui signifie «ou». Avec `grep`, il faut backslasher le pipe, avec `egrep` ce n'est pas nécessaire. On écrit donc (avec `grep`) :

```
grep '^.r\|[[:space:]].r' fichier
```

Ce n'est quand même pas si simple; les mots peuvent être précédés d'un tiret (mots composés), d'une apostrophe, de guillemets divers (``, ", «, <<), et, si l'auteur du texte n'est pas respectueux des règles de typographie, d'une ponctuation. Il y a donc bien des cas à envisager...

```
grep '\<[a-zA-Z]*[A-Z]r' fichier

grep '^[A-Z]r\|[:space:][a-zA-Z]*[A-Z]r' fichier
```