

TP n°1

Mise en place des données

Récupérer le fichier archive nommé lte2_files.tgz situé à l'adresse suivante :
http://www.info.univ-angers.fr/pub/basseur/lte2_files.tgz

Décompressez l'archive puis la supprimer en tapant successivement les commandes :

```
tar -xvzf lte2_files.tgz
rm lte2_files.tgz
```

Exercice 1:

Déplacez vous dans le répertoire dir3.
 Listez le contenu de ce répertoire. Combien y à t'il de fichiers ?

Avant d'écrire les commandes suivantes, essayez d'évaluer le nombre de fichiers qui seront affichés en exécutant les commandes suivantes. Exécutez ensuite ces commandes pour vérifier.

Commande	#fichiers	Commande	#fichiers	Commande	#fichiers
ls a*		ls annee*		ls *a*e*	
ls A*		ls annee?		ls *e*a*	
ls *c*		ls archive*		ls *??-03	
ls *e		ls archive*-*		ls *[0]*	
ls * *		ls archive??-02		ls *2?*	
ls *		ls ar*-*		ls *2?	
ls ?		ls [aA]nnee*		ls *[3-9]	
ls ?nn*		ls ar*[12]?-*		ls *[v-z]*	

Remarque : l'utilisation des « * », « ? » et « [] » est la même pour toutes les commandes unix.

Exercice 2:

Créer le répertoire Year dans le répertoire courant, en une commande déplacez tout les fichiers exceptés les fichiers archives dans ce nouveau répertoire. Vérifiez le résultats en affichant le contenu du répertoire Year sans vous déplacer dans ce répertoire.

Exercice 3:

Lister tout les fichiers :

- ◆ Se terminant par 5,
- ◆ commençant par annee4,
- ◆ commençant par annee4 et de 7 lettres exactement,
- ◆ commençant par annee suivi d'aucun chiffre numérique,
- ◆ contenant la chaîne ana,
- ◆ commençant par a ou A.

Exercice 4:

Copier les fichiers dont l'avant dernier caractère est un 4 ou 1 dans le sous-répertoire **tmp** (créé préalablement) en une seule commande.

Les droits d'accès

Exercice 1:

Dans votre répertoire courant, créez un répertoire courant **essai_droit**, par défaut ce répertoire est à 755 (rwxr-xr-x), quelles sont les commandes (en notation symbolique et en base 8) pour lui donner les droits suivant (on suppose qu'après chaque commande on remet le répertoire à 755):

	propriétaire			groupe			les autres		
	droit en lecture	droit en écriture	droit d'accès	droit en lecture	droit en écriture	droit d'accès	droit en lecture	droit en écriture	droit d'accès
commande 1	oui	oui	oui	oui	non	oui	non	non	oui
commande 2	oui	non	oui	non	oui	non	non	non	oui
commande 3	non	oui	non	non	non	oui	oui	non	non
commande 4	non	non	oui	oui	non	oui	non	non	non

Exercice 2:

Créez un fichier **droit** dans le répertoire **essai_droit**, par défaut ce fichier est à 644 (rw-r-r-). En partant du répertoire courant, pour chaque commande de l'exercice précédent, essayez d'accéder au répertoire **essai_droit** (commande `cd`), de faire un `ls` dans **essai_droit** et de modifier le fichier avec un éditeur quelconque (emacs par exemple).

Grep

Exercice 1

Créer un répertoire **essai-grep** dans votre home directory. La commande `ls > essai.txt` permet de lister les fichiers du répertoire courant et de rediriger le résultat dans le fichier `essai.txt`.

Editez les fichiers (sortie de la commande `ls` redirigée vers `grep`) avec les critères sur leur nom suivant:

- Le nom doit commencer par `Annee` ou `annee`
- `01` est en fin de nom
- `iv` est présent dans le nom
- Nom contenant au moins un chiffre numérique
- Nom contenant la chaîne `nn` ou `11`

Exercice : Séquence d'ADN.

Déplacez vous dans le répertoire `dir4`.

Les fichiers `seqs?.txt` contiennent des séquences ADN que l'on veut analyser, à l'aide de la commande `grep`.

Voici d'abord quelques informations concernant la commande `grep` (mais vous pouvez aussi utiliser la commande `man grep` pour plus d'informations !)

Syntaxe : `grep [options] expreg [fichiers]`

Cette commande recherche dans les fichiers ou sur son entrée standard des lignes de texte qui satisfont l'expression régulière `expreg` indiquée. Sa sortie peut être redirigée dans un fichier

Options

- c donne seulement le nombre de lignes trouvées obéissant au critère
- l donne seulement le nom des fichiers où le critère a été trouvé
- v donne les lignes où le critère **n'a pas** été trouvé
- i ne pas tenir compte de la casse (ne pas différencier majuscules minuscules)
- n pour n'afficher que les numéros des lignes trouvées
- w pour imposer que le motif corresponde à un mot entier d'une ligne

Expressions reconnues

Voici la liste des symboles utilisables par `grep` : `.` `*` `[]` `^` `$`

- `.` signifie un caractère quelconque
- `*` répétition du caractère situé devant
- `^` début de ligne
- `$` fin d'une ligne (donc `"e$"` mots se terminant par e)
- `[...]` contient une liste ou un intervalle de caractères cherchés
- `[^..]` caractères interdits.
- ...et ce n'est pas fini !!!

Attention

Pour éviter une confusion entre les interprétations de ces symboles spéciaux par `grep` ou par le **shell**, il est indispensable de "verrouiller" `expreg` en plaçant l'expression entre guillemets (et non entre quotes !).

Réaliser les manipulations suivantes :

1. Afficher les séquences du fichier `seqs1.txt` contenant la séquence « ACGT ».
2. Même question, mais en redirigeant le résultat dans un fichier `2.txt`.
3. Calculer le nombre de séquences contenant 6 « C » consécutifs au moins.
 - En recherchant la chaîne « CCCCCC »
 - En utilisant une autre méthode (utilisez `man` pour trouver comment faire!)
4. Chercher dans ce fichier, les lignes dont la 1ère lettre est quelconque et la 2ème doit être C.
5. Chercher les lignes ne commençant pas commençant par « G »
6. Chercher les lignes contenant un « GC » suivi d'un caractère quelconque, itéré deux fois de suite (exemple `ACCTGCAGCCTT`).
7. Compter le nombre de lignes se terminant par « A ».
8. Tester si l'un des fichiers `seqs?.txt` contient un caractère erroné (autre que A, C, G ou T).