

Exercice 1 : Fichiers

Q 1. Écrire un script, nommé `contenu-fichiers`, qui affiche le contenu de tous les fichiers dont les noms lui sont passés en paramètre, en précisant le nom de chaque fichier avant l'affichage de son contenu.

Q 2. Écrire un script, nommé `taille-fichiers`, qui affiche pour chaque fichier du répertoire courant une ligne de la forme `<nom_fichier> taille`.

Q 3. Écrire un script, nommé `vieillir-fichiers`, qui prend en argument des chemins de fichiers. Une option `-v` peut également être placée n'importe où parmi les arguments. Si l'option est présente, la commande affiche les caractéristiques de chaque fichier et répertoire (au même format que l'affichage fait par la commande `ls -l` mais sans afficher le contenu des répertoires). Si l'option n'est pas présente, la commande renomme chacun des fichiers (ou répertoires) en ajoutant le suffixe `.old`.

Q 4. Écrire un script, nommé `archiver-fichiers`, qui surnomme tous les fichiers de suffixe `.txt` du répertoire courant en des fichiers de même nom de base mais de suffixe `.txt.archive`. Si l'option `--backup-dir` est présente le surnommage doit se faire dans le répertoire dont le nom suit l'option, sinon le surnommage se fait dans le répertoire courant.

Q 5. Écrire une commande, nommé `jeter-fichiers`, qui permet de manipuler une poubelle de fichiers. Ce sera un simple répertoire nommé `poubelle` située dans la racine de votre espace personnel. La commande a trois options :

- `jeter -l` pour lister le contenu de la poubelle ;
- `jeter -r` pour vider la poubelle ;
- `jeter fichier1 fichier2 ...` pour déplacer les fichiers considérés vers la poubelle.

Si la poubelle n'existe pas, elle est créée à l'appel de la commande.

Q 6. Écrire une commande, nommé `lister-executables`, qui prend en argument un nombre quelconque de noms de fichier et affiche pour chacun :

- son nom si c'est un fichier régulier exécutable ;
- la liste des fichiers exécutables qu'il contient si c'est un répertoire.

Exercice 2 : Arithmétique

Q 1. Écrire un script, nommé `somme`, qui prend un nombre indéterminé de paramètres et calcule leur somme.

Q 2. Écrire un script, nommé `calculs`, qui boucle en demandant à l'utilisateur de saisir deux variables, affiche leur somme, leur différence, leur produit et leur quotient. Le programme s'arrête lorsque l'utilisateur saisit un point.

Q 3. Écrire un script, nommé `factorielle`, qui prend un argument et en affiche la factorielle. Vérifier que le nombre de paramètres est correct et que c'est bien un nombre entier.