1 Révision : commande de base

Pour chacune des question suivantes donner une commande (éventuellement composée de plusieurs commandes de base reliées par des tubes) qui permet de réaliser la tâche demandée.
— Créer un fichier de taille 0 octet nommé liste. Ajoutez à ce fichier la liste des fichiers contenus dans votre répertoire de connexion. Afficher l'avant deux dernières lignes du fichier liste.

```
> liste                 # créer un fichier vide
ls ~ > liste            # liste les fichiers du répertoire de connexion et les met dans liste
tail -n 2 liste         # afficher les 2 dernières lignes du fichier
```

— Expliquer la différence entre les deux commandes echo * et ls *.

echo *
ls *
`echo *` → `fichier1 fichier2 dossier1`
`ls *` → affiche le contenu de `fichier1`, `fichier2`, et la liste des fichiers dans `dossier 1`.

— Donner une commande qui permet d'effacer tous les fichiers de taille 0 octet se trouvant dans le répertoire /tmp.

find /tmp -type f -size 0 -delete
find /tmp → cherche dans /tmp

-type f → seulement les fichiers

-size 0 → fichiers de taille 0 octet

-delete → supprime les fichiers trouvés

— Le fichier /etc/hosts donne une liste d'association entre adresses IP et des noms symboliques de machines. Donner une commande qui renvoie l'adresse IP associée à un nom symbolique.

grep -w "localhost" /etc/hosts | awk '{print $1}'
grep -w → cherche le mot exact

awk '{print $1}' → affiche la première colonne (l'adresse IP)

— Le fichier /etc/services précise pour chaque service réseau le type et le numéro de port à utiliser. Donner une commande qui renvoie le numéro de port si on fournit le nom du service et le type du protocole de transport à utiliser (TCP ou UDP).

grep -w "ssh" /etc/services | grep tcp | awk '{print $2}' | cut -d/ -f1
grep -w "ssh" → cherche la ligne correspondant au service

grep tcp → filtre pour le protocole TCP

awk '{print $2}' → prend la deuxième colonne (ex : 22/tcp)

cut -d/ -f1 → extrait seulement le numéro de port (avant le /)

2 Variable d'environnement

```
X1=3
Y1=10
Z1=4
export Y1
env | grep X1=
echo $X1
echo $x1
env | grep Y1=
unset Y1
export X1
bash
env | grep X1=
echo $Z1
exit
echo $Z1
```

```
esteban_derisbourg@j009-06:~$ X1=3
esteban_derisbourg@j009-06:~$ Y1=10
esteban_derisbourg@j009-06:~$ Z1 = 4
Z1 : commande introuvable
esteban_derisbourg@j009-06:~$ Z1=4
esteban_derisbourg@j009-06:~$ export Y1=10
esteban_derisbourg@j009-06:~$ env | grep X1=
esteban_derisbourg@j009-06:~$ echo $X1
3
esteban_derisbourg@j009-06:~$ echo $x1

esteban_derisbourg@j009-06:~$ env | grep Y1=
Y1=10
esteban_derisbourg@j009-06:~$ unset Y1
esteban_derisbourg@j009-06:~$ export X1
esteban_derisbourg@j009-06:~$ bash
esteban_derisbourg@j009-06:~$ env | grep X1=
X1=3
esteban_derisbourg@j009-06:~$ echo $Z1

esteban_derisbourg@j009-06:~$ exit
exit
esteban_derisbourg@j009-06:~$ echo $Z1
4
esteban_derisbourg@j009-06:~$
```

3 Scripts bash

Écrire un script bash ldir permettant d'afficher le contenu d'un répertoire en séparant les fichiers et les (sous)répertoires. Exemple d'utilisation : ldir /etc affiche : Fichiers dans /etc : Répertoires dans /etc :

```
W#!/bin/bash

echo ""
echo "Fichiers dans $1 :"
echo ""
find "$1" -maxdepth 1 -type f -printf "%f\n" | sort
echo ""
echo ""
echo "Répertoires dans $1 :"
echo ""
find "$1" -maxdepth 1 -type d ! -path "$1" -printf "%f\n" | sort
```

find "$DIR"
Cherche dans le répertoire $DIR.
-maxdepth 1
Ne cherche que dans ce répertoire, sans aller dans les sous-dossiers.
-type f
Ne sélectionne que les fichiers (pas les répertoires ni autres types).

-printf "%f\n"
Affiche seulement le nom du fichier (sans tout le chemin complet).

%f = nom du fichier seul
\n = saut de ligne, pour que chaque fichier soit sur une ligne différente.

| sort
Trie la liste de fichiers par ordre alphabétique.

Écrire un programme shell nommé sauvegardeTxt :

```
#!/bin/bash
# Répertoire personnel
SOURCE_DIR="$HOME"
# Répertoire de sauvegarde caché
BACKUP_DIR="$HOME/.BACKUP"
# Crée le répertoire .BACKUP s'il n'existe pas
if [ ! -d "$BACKUP_DIR" ]; then
    mkdir "$BACKUP_DIR"
    echo "Création du répertoire $BACKUP_DIR"
fi
# Recherche tous les fichiers .txt dans $SOURCE_DIR et sous-répertoires
find "$SOURCE_DIR" -type f -name "*.txt" | while read -r fichier; do
    nom_fichier=$(basename "$fichier")

    # Vérifie si le fichier existe déjà dans .BACKUP
    if [ -e "$BACKUP_DIR/$nom_fichier" ]; then
        echo "Existant, non copié : $nom_fichier"
    else
        cp "$fichier" "$BACKUP_DIR/"
        echo "Copié : $nom_fichier"
    fi
done
```

Développer une commande nommée poubelle qui permet de transférer les fichiers à effacer dans un répertoire nommé trash.

La syntaxe de cette commande est la suivante : poubelle fichier a pour effet de transférer le fichier dans le répertoire trash. L'appel de la commande sans arguments a pour effet d'afficher un message d'aide décrivant la syntaxe correcte de la commande.

```bash
TRASH_DIR="$HOME/trash"


afficher_aide() {
    echo "Usage : poubelle <fichier1> [fichier2 ...]"
    echo "Déplace le(s) fichier(s) spécifié(s) dans le répertoire trash."
    echo "Le répertoire trash est situé dans : $TRASH_DIR"
}

# Si aucun argument n'est fourni
if [ $# -eq 0 ]; then
    afficher_aide
    exit 1
fi

# Création du répertoire trash si nécessaire
if [ ! -d "$TRASH_DIR" ]; then
    mkdir -p "$TRASH_DIR"
fi

# Pour chaque fichier passé en argument
for fichier in "$@"; do
    if [ -e "$fichier" ]; then
        mv "$fichier" "$TRASH_DIR"/
        if [ $? -eq 0 ]; then
            echo "✅ $fichier déplacé dans $TRASH_DIR/"
        else
            echo "❌ Erreur : impossible de déplacer $fichier"
        fi
    else
        echo "⚠️ Fichier inexistant : $fichier"
    fi
done
```

Random

```bash
RGB_FILE="/etc/X11/rgb.txt"
if [ ! -f "$RGB_FILE" ]; then
        echo "Erreur : fichier $RGB_FILE introuvable."
        exit 1
fi

colors=($(awk '!/^#/ && NF>=4 { $1=$2=$3=""; sub(/^[ \t]+/, ""); print }' "$RGB_FILE"))

nb_colors=${#colors[@]}

if [ "$nb_colors" -eq 0 ]; then
        echo "Aucune couleur trouvée dans $RGB_FILE."
        exit 1
fi

index=$(( RANDOM % nb_colors ))

color="${colors[$index]}"

echo "Lancement de xterm avec la couleur de fond : $color"

xterm -bg "$color"
```

Point bonus : quand tu te connectes au pc, terminal s'ouvre "bienvenu mouusieur... si vous n'êtes pas mouusieur... déconnectez"