

TP Programmation Shell

Exercice 1 :

Ecrire un programme **verifIdentité** qui détermine si le paramètre passé en argument est un fichier ou un répertoire.

```
#!/bin/bash

if [ $# = 0 ]; then
    echo "Entrez un paramètre"
elif [ ! -e "$1" ]; then
    echo "$1 n'existe pas"
elif [ -f "$1" ]; then
    echo "$1 est un fichier"
elif [ -d "$1" ]; then
    echo "$1 est un repertoire"
fi
```

Exercice 2 :

Ecrire un programme **afficheFormulaire** qui demande de saisir une liste de données comme : le nom, le prénom, l'âge et l'adresse, ensuite permet de les afficher.

```
#!/bin/bash

read -p "Entrez votre nom: " Nom
read -p "Entrez votre prénom: " Prenom
read -p "Entrez votre age: " Age
read -p "Entrez votre adresse: " Adress

echo "Salut $Nom $Prenom, vous avez $Age ans, Vous habitez à $Adress"
```

Exercice 3 :

Ecrire un programme **copieArchiveHome** qui prend en argument le dossier personnel de l'utilisateur et les copie dans un dossier nommé ARCHIVES-NOMUSER.

```
#!/bin/bash

if [ $# = 0 ]; then
    echo "Entrez le nom du répertoire personnel"
elif [ ! -e $1 -o ! -d $1 ]; then
    echo "Ce répertoire n'existe pas"
elif [ -d $1 -a $1 != $HOME ]; then
    echo "Ce répertoire personnel n'existe pas"
else
    User=`whoami`
    sudo cp -r $1 /ARCHIVE-$User
    echo "ssalut"
fi
```

Exercice 4 :

Ecrire un programme **effacerFichier** qui détruit tous les fichiers passés en paramètre, mais, avant de l'effacer, le programme montre les 5 premières lignes et demande une confirmation pour effacement.

```
#!/bin/bash

if [ $# = 0 ]; then
    echo "Entrez les fichiers"
else
    for var in "$@"
    do
        if [ -e "$var" ]; then
            if [ -f "$var" ]; then
                rm -i "$var"
            else echo "ce n'est pas un fichier"
            fi
        else echo "ce fichier n'existe pas"
        fi
    done
fi
```

Exercice 5 :

Ecrire un programme **ajouteDateEntete** qui ajoute à l'entête du fichier la date actuelle.

```
#!/bin/bash

if [ $# = 0 ]; then
echo "Entrez un nom de fichier"
elif [ -e $1 ] && [ -f $1 ]; then
sed -i "li $(date)" $1
else echo "ce fichier n'existe pas"
fi
```

Exercice 6 :

Ecrire un programme **fichRep** qui prend en argument le nom d'un répertoire et qui liste les fichiers de ce répertoire.

```
#!/bin/bash

if [ $# = 0 ]; then
echo "Entrez un nom de repertoire"
elif [ -e $1 ] && [ -d $1 ]; then
ls -l $1 | grep ^- | rev | cut -d ' ' -f 1 | rev
else echo "ce repertoire n'existe pas"
fi
```

```
#!/bin/bash

if [ $# = 0 ]; then
echo "Entrez un nom de repertoire"
elif [ -e $1 ] && [ -d $1 ]; then
find $1 -type f -maxdepth 1
else echo "ce repertoire n'existe pas"
fi
```

Exercice 7 :

Ecrire un programme **nbreDossFich** prenant en argument un répertoire quelconque et qui affiche le nombre de dossier et fichiers qui existent dans ce dernier.

```
#!/bin/bash

if [ $# = 0 ]; then
echo "Entrez un nom de repertoire"
elif [ -e $1 ] && [ -d $1 ]; then
rep=`ls -l $1 | grep ^d | wc -l`
file=`ls -l $1 | grep ^- | wc -l`
echo "ce repertoire contient $rep repertoire et $file fihiers"
else echo "ce repertoire n'existe pas"
fi
```

Exercice 8 :

Ecrire un programme **observations** prenant en argument une série de moyennes de matières et attribue pour chaque note une observation, exemple, le programme doit afficher le texte suivant :

Il y a **nbre** moyennes qui sont introduites.

Il y a **nbre** notes qui sont : assez bien.

Il y a **nbre** notes qui sont : bien.

etc ...

```

#!/bin/bash

Excellent(){
test $1 -ge 18 && return 0 || return 1
}

Tres_bien(){
test $1 -ge 16 -a $1 -lt 18 && return 0 || return 1
}

Assez_bien(){
test $1 -ge 14 -a $1 -lt 16 && return 0 || return 1
}

Bien(){
if [ $1 -ge 12 -a $1 -lt 14 ]; then
return 0
else
return 1
fi
}

Moyen(){
if [ $1 -ge 10 -a $1 -lt 12 ]; then
return 0
else
return 1
fi
}

Faible(){
if [ $1 -lt 10 ]; then
return 0
else
return 1
fi
}

if [ $# = 0 ]; then
echo "Entrez les moyennes"
else
e=0
t=0
a=0
b=0
m=0
f=0
for var in "$@"
do
if Excellent "$var"; then let e=e+1
fi
if Tres_bien "$var"; then
let t=t+1
fi
if Assez_bien "$var"; then
let a=a+1
fi
if Bien "$var"; then
let b=b+1
fi
if Moyen "$var"; then
let m=m+1
fi
if Faible "$var"; then
let f=f+1
fi
done
echo " Il y'a $# moyennes qui sont introduites:
Il y'a $e notes qui sont: Excellent
Il y'a $t notes qui sont: Très bien
Il y'a $a notes qui sont: Assez bien
Il y'a $b notes qui sont: Bien
Il y'a $m notes qui sont: Moyen
Il y'a $f notes qui sont: Faible "
fi

```