



TPN°5 : Les tests et conditions

- Créer des scripts Shell avec des tests : **if...then** et **Case ...esac**

Exercices :

1. Écrire un script qui lit un entier à partir du clavier puis affiche son signe (négatif, positif ou nul).

```
echo -n " Donner un entier : "  
read n  
if test $n -gt 0  
then echo "le signe de $n est +"  
elif test $n -lt 0  
then echo "le signe de $n est -"  
else  
echo " $n est nulle"  
fi
```

2. Écrire un script qui lit 2 chaînes de caractères comme paramètres puis compare leurs valeurs et affiche le résultat sous forme d'un message.

Exemple d'exécution : **bash script2.2.sh ch1 ch2**

ch1 et ch2 sont différentes

```
if test $# -eq 2 # tester si le nombre d'arguments est 2  
then  
    if test $1 == $2  
    then echo "$1 et $2 sont identiques"  
    else  
        echo "$1 et $2 sont différentes"  
    fi  
else  
echo " Le nombre d'argument incorrect !"  
fi
```



3. Écrire un script qui affiche le menu suivant : 1 – Windows? 2 – Kalinux? 3 – Linux? 4 – Unix? Réponse ? Si vous répondez 1 alors le programme affiche « Dommage ! », 2 il affiche « Peut mieux faire ! », 3 « Pas mal ! », 4 « Super! ».

```
echo "  
1 – Windows ?  
2 – Kalinux ?  
3 – Linux ?  
4 – Unix ?  
Votre Réponse ? "  
read R  
case $R in  
"1") echo "Dommage";;  
"2") echo "Peut mieux faire !";;  
"3") echo "Pas mal ";;  
"4") echo " Super !";;  
*) echo " Mauvaise réponse"  
esac  
echo " Fin de l'exercice "
```

4. Écrire un script qui dit si le premier paramètre passé en ligne de commande est un fichier, un répertoire, ou autre type. Exemple d'exécution : **uic@localhost:~\$ bash script2.4.sh /etc** /etc est un dossier **uic@localhost:~\$ bash script2.4.sh /etc/passwd** /etc/passwd est un fichier **uic@localhost: ~\$ bash script2.4.sh /dev/null** /dev/null est un autre type de fichier



```
if test $# -eq 1
then
    if test -f $1
    then echo "$1 est un fichier ordinaire"
    elif test -d $1
    then echo " $1 est un dossier"
    else echo " $1 est un autre type "
    fi
else
    echo " Nombre d'argument incorrect !"
fi
```

5. Écrire un script qui concatène puis trie deux fichiers file1 et file2 dans un nouveau fichier file3 et qui affiche le nombre total de lignes. Les noms des trois fichiers doivent être passés en paramètre.

```
if test $# -eq 3
then
    if test -f $1 ;then
        if test -f $2; then
            cat $1 $2 > $3
            sort $3
            echo -n " le nombre de lignes de $3 est " ;
            wc -l $3
        fi
    fi
else
    echo " nombre d'arguments incorrect !"
fi
```

6. Modifier le script précédent pour demander à l'utilisateur de saisir au clavier le (ou les) nom(s) de fichiers qu'il aurait oublié



d'indiquer en lançant le script

7. Écrire une commande recycle qui permet de manipuler une corbeille de fichiers (un répertoire) nommée corbeille et située à votre répertoire personnel. La commande accepte trois options : **recycle -l pour lister le contenu de la corbeille;**

recycle -r pour vider la corbeille ;

recycle fichier1 fichier2 ... pour déplacer les fichiers considérés vers la corbeille. Si la corbeille n'existe pas, elle est créée à l'appel de la commande.

Exemple de la boucle While

```
echo "Tapez le mot de passe :"  
read password  
while test "$password" != "1234"  
do  
echo "Mot de passe incorrect."  
echo "Tapez le mot de passe"  
read password  
done  
echo "Mot de passe correct."  
echo "Bienvenue !"
```



Exercice 1 Affichez les 10 premiers multiples d'un entier saisi au clavier

```
read -p "Entrer un entier " n
i=1
while test $i -le 10
do
echo ${n*i}
i=$((i+1))
done
```

Calculez n ! (n est un entier saisi au clavier)

```
read -p "Entrer un entier n " n
i=1
f=1
while test $i -le $n
do
    f=$((f*i))
    i=$((i+1))
done
echo $n! = $f
```



Écrire un script qui compte le nombre de fichiers et de répertoires dans le répertoire courant

```
# Écrire un script qui compte le nombre de fichiers  
# et de répertoires dans le répertoire courant  
  
nb_f=0  
nb_d=0  
for fich in *  
do  
    if test -f $fich  
    then nb_f=${nb_f+1}  
    elif test -d $fich  
    then nb_d=${nb_d+1}  
    fi  
done  
  
echo " Le nombre de fichiers est $nb_f"  
echo " Le nombre de dossiers est $nb_d"
```



**# Écrire un script Shell qui donne pour un répertoire
donné en argument le nombre de chacun des types
des fichiers suivant : Fichiers exécutables,
fichiers accessibles en lecture et fichiers
accessibles en écriture**

```
nb_x=0 # exécutable
nb_w=0 # accessible en écriture
nb_r=0 # accessible en lecture
for fich in *
do
    if test -x $fich
    then nb_x=${nb_x+1}
    fi
    if test -w $fich
    then nb_w=${nb_w+1}
    fi
    if test -r $fich
    then nb_r=${nb_r+1}
    fi
done

echo " Le nombre de fichiers exécutables $nb_x"
echo " Le nombre de fichiers accessibles en lecture $nb_r"
echo " Le nombre de fichiers accessibles en écriture $nb_w"
```