

TP2

Exercice 1

1. Bash trouve les commandes tapées par l'utilisateur dans les dossiers pointés par les différents chemins absolus stockés dans la variable **PATH**
2. La commande **cd** utilise la variable d'environnement **HOME** contenant l'emplacement du répertoire personnel de l'utilisateur actuel
3. La variable **LANG** contient le paramètre linguistique de base utilisé par les applications du système. La variable **PWD** contient le répertoire de travail courant du terminal de commande. La variable **OLDPWD** contient le répertoire de travail précédant le dernier appel de la commande **cd** La variable **SHELL** contient l'interpréteur de commande préféré de l'utilisateur tel qu'il est défini dans le fichier « /etc/passwd » . La variable **_** contient la dernière commande tapée dans bash. En tapant **\$_** on peut re-exécuter cette commande.
4. On crée la variable avec la commande **export VAR="variable"** et on la visualise avec **printenv VAR**
5. La variable créée n'est pas reconnue en mode root car sa création est temporaire et effective uniquement pour la sessions courante lors de sa création.
6. En initialisant la variable dans le fichier **.bashrc** (depuis le repertoire personnel **~/**) on rend la création/modification d'une variable permanente. En effet ce fichier est lu a chaque démarrage du bash.
7. On initialise la variable NOMS dans **.bashrc** et on exécute la commande **source .bashrc** pour forcer bash a relire le fichier **.bashrc** (reboot du bash)
8. On tape la ligne de commande **echo "Bonjour vous deux, \$NOMS !"**
9. **unset** permet de supprimer la référence a la variable alors qu'une variable vide a une référence mais qui ne pointe vers rien.
10. On tape la ligne de commande **echo "\$HOME = \$HOME"**

Exercice 2

Code de testpwd.sh: `#!/bin/bash export pwd="password" echo "Saisie mot de passe:" read -s mdp if [$mdp = $pwd]; then echo "Ok" else echo "Mot de passe faux!" fi`

Exercice 3

Code de rationnel.sh: `#!/bin/bash function is_number() { re='^[+-]?[0-9]+([.][0-9]+)?$' if ! [[$1 =~ $re]] ; then return 1 else return 0 fi }`

`is_number $1 if [$? = 0]; then echo "Nombre reel" else echo "Type non reel" fi`

Exercice 4

Code de users.sh: `#!/bin/bash`

```
function verifyUser() { res=grep $1 /etc/passwd echo $res if [ $(#res) -ne 0 ]; then echo "L'utilisateur existe"
return 1 fi echo "L'utilisateur n'existe pas" return 0 }
```

```
if [ -z $1 ]; then echo "Utilisation de : " $0 ": cmd + nom-utilisateur" exit fi verifyUser $1
```

Exercice 5

Code de factoriel.sh: `#!/bin/bash export res=1 for ((i = 1; i <= $1; i++)) do res=$((res * $i)) done echo $res`

Exercice 6

Code de justePrix.sh: `#!/bin/bash num=$RANDOM RANGE=100 let "num %= $RANGE" guess=$1`

```
function justePrix() { if [ $guess -lt $num ]; then echo "C'est plus!" read guess justePrix elif [ $num -lt $guess ];
then echo "C'est moins!" read guess justePrix else echo "C'est gagné!" return 1 fi }
```

```
if [ -z $1 ]; then echo "Utilisation de : " $0 ": cmd + numero" exit fi justePrix $guess
```

Exercice 7

Q1 :

Code de statistiques.sh: `#!/bin/bash`

```
min=100 max=-100 moy=0
```

```
function is_number() { re='^[+-]?[0-9]+([.][0-9]+)?$' if [[ $1 =~ $re ]]; then if [ $1 -le 100 ]; then if [ $1 -ge -100
]; then return 1 fi fi fi echo "erreur: saisissez 3 entiers entre -100 et 100" exit }
```

```
function minMax() { if [ $1 -lt $min ]; then min=$1 elif [ $1 -gt $max ]; then max=$1 fi return 1 }
```

```
function moyenne() { moy=$((($1 + $2 + $3) / 3)) return 1 }
```

```
is_number $1 is_number $2 is_number $3
```

```
minMax $1 minMax $2 minMax $3
```

```
echo "min : $min" echo "max : $max"
```

```
moyenne $1 $2 $3
```

```
echo "moyenne : $moy"
```

Q2 :

```
#!/bin/bash
```

```
min=100 max=-100 moy=0 nb_args=$#
```

```
function somme() { moy=$((moy + $1)) return 1 }
```

```
function moyenne() { moy=$((moy / $nb_args)) return 1 }
```

```
for i in $(seq 1 $#) do is_number $1 minMax $1 somme $1 shift done
```

```
moyenne
```

```
echo "min : $min" echo "max : $max" echo "moyenne : $moy"
```

Q3 :

```
#!/bin/bash
```

```
min=100 max=-100 moy=0 nb_notes=1
```

```
function is_number() { re='^[+-]?[0-9]+([.][0-9]+)?$' if [[ $1 =~ $re ]] ; then if [ $1 -le 100 ] ; then if [ $1 -ge -100 ] ; then return 1 fi fi fi echo "erreur: saisissez des entiers entre -100 et 100" exit }
```

```
function minMax() { if [ $1 -lt $min ] ; then min=$1 elif [ $1 -gt $max ] ; then max=$1 fi return 1 }
```

```
function somme() { moy=$((moy + $1)) return 1 }
```

```
function moyenne() { moy=$((moy / $nb_notes)) return 1 }
```

```
for line in $(cat note.txt) do nb_notes=$((nb_notes + 1)) is_number $line minMax $line somme $line shift done
```

```
moyenne
```

```
echo "Fichier lu : note.txt" echo "min : $min" echo "max : $max" echo "moyenne : $moy"
```