

# TP7- Programmation Shell

## Correction

### Exercice 1

```
#!/bin/bash
echo $(date -u)
cal
```

### Exercice 2

1.

```
#!/bin/bash
echo "Entrer le nom d'un répertoire : "
read dir
echo "Le répertoire $dir contient les fichiers suivants : "
ls $dir
```

2.

```
#!/bin/bash

test -f $1 && echo fichier présent
test -f $1 || echo fichier non présent
```

3.

```
#!/bin/bash

test -d $1 && echo répertoire présent
test -d $1 || echo répertoire non présent
```

### Exercice3

```
#!/bin/bash

echo Il y a $# arguments

test $# -ge 1 && echo Le premier argument est: $1
```

### Exercice4

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

head -3 \$1 | tail +\$2

### Exercice5

```
#!/bin/bash

echo "Entrez votre note :"read -r note

if [ "$note" -ge 16 ]; then
    echo "très bien"
elif [ "$note" -ge 14 ]; then
    echo "bien"
elif [ "$note" -ge 12 ]; then
    echo "assez bien"
elif [ "$note" -ge 10 ]; then
    echo "moyen"
else
    echo "insuffisant"
fi
```

### Exercice6

```
#!/bin/bash
i=1
while test $i -le $1; do
    echo $i;
    let i=$i+1;
done
```

### Exercice7

1.

```
#!/bin/bash

for fichier in .//* do

    echo "$fichier"

done
```

2.

```
#!/bin/bash

for par in $* do
```

```
echo $par
```

```
done
```

## Exercice8

```
#!/bin/bash
# script test-fichier
if [ $# -ne 1 ]; then
    echo "Usage : $0 " exit 1
fi
if [ ! -e "$1" ]; then
    echo "Le fichier '$1' n'existe pas" exit
fi
if [ ! -s "$1" ]; then
    echo "$1 a une taille nulle"
exit
fi
# droits et type d'utilisateur
a=$(stat -c%A "$1")
p=$(stat -c%U "$1")
g=$(stat -c%G "$1")
n=1 echo "'$1' est accessible par :"
for u in "le propriétaire ($p)" "le groupe ($g)" "les autres"
do
    echo -n "* $u en"
    d=${a:n:3}
    if [ $d == --- ] then
        echo -n " rien"
    else
        [[ $d == r?? ]] && echo -n " lecture"
        [[ $d == ?w? ]] && echo -n " écriture"
        [[ $d == ??x ]] && echo -n " exécution"
    fi
    fi
    n=$((n+3)) echo
done
# type de fichier
case ${a::1} in
    -) echo "fichier standard" ;;
    d) echo "dossier" ;;
    l) echo "lien symbolique" ;;
esac
```

## Exercice 9

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

```

if [ "$#" -eq 0 ]; then
    echo "Saisir une valeur : "
    read -r val
else
    val=$1
fi
# Dans le cas où c'est négatif, on rend la valeur positive
if [ "$val" -lt 0 ]; then
    let val=-1*$val
fi
result=1
val2="$val"
while [ "$val" -ne 0 ]; do
    printf "$val "
    let result=$result*$val
    let val=$val-1
    if [ "$val" -ne 0 ]; then
        printf "*"
    fi
done
echo "= $result

```

## Exercice 10

```

#!/bin/bash
somme=0
while :
do
    echo -e "Saisir un nombre entier : \c"
    if read nombre
    then
        if ( expr "$nombre" : '[0-9]*$' || expr "$nombre" : '-[0-9]*$' ) > /dev/null
        then
            somme=`expr $somme + $nombre`
        else
            echo "Saisie incorrecte"
        fi
    else
        break
    fi
done
echo -e "\nResultat : $somme"
exit 0

```