



# *TP programmation sous Shell*

Compte rendu

**Préparé par : Fatima ezzhrae Maamouch**

## Exercice 1 :

écrire un programme calculant le nième terme  $U_n$  de la suite de FIBONACCI donnée par la relation de récurrence:

**$U_1=1$   $U_2=2$  et  $U_n=U_{n-1}+U_{n-2}$  (pour  $n>2$ )**

```
#!/bin/bash|
u1=1
u2=1
echo "entrer un nombre "
read n
for i in $(seq 2 $n)
do
  Un=$(( $u1 + $u2 ))
  u1=$u2
  u2=$Un
done
echo "la solution est : U$n=$u2"
```

```
fz@fedora:~
[fz@fedora ~]$ ./script.sh
entrer un nombre
3
la solution est : U3=3
[fz@fedora ~]$
```

## Exercice 2:

Dans un cinéma, on désire calculer la moyenne d'âges de personnes s'intéressant à voir un film particulier. Pour finir, l'utilisateur doit répondre par 'N' à la question posée par le programme

"Encore une autre personne (O/N)?"

Et on doit afficher la moyenne d'âges des personnes à l'écran.

```
#!/bin/bash
echo " entrer une réponse"
read reponse
s=0
i=0
while [ $reponse -eq 1 ]
do
i=$(( $i +1))
echo "entrer l'age d'une personne num $i"
read age
s=$(( $s + $age))
echo " encore une autre personne ? taper 1 pour oui et taper 0 pour non "
read reponse
done
m=$(( $s / $i))
echo "la moyenne d'age des personnes est : $m"
```

```
[fz@fedora ~]$ ./script.sh
 entrer une réponse
1
entrer l'age d'une personne num 1
10
encore une autre personne ? taper 1 pour oui et taper 0 pour non
1
entrer l'age d'une personne num 2
12
encore une autre personne ? taper 1 pour oui et taper 0 pour non
1
entrer l'age d'une personne num 3
10
encore une autre personne ? taper 1 pour oui et taper 0 pour non
0
la moyenne d'age des personnes est : 10
[fz@fedora ~]$
```

### Exercice 3:

Les habitants paient l'impôt selon les règles suivantes :

les hommes de plus de 20 ans paient l'impôt

les femmes paient l'impôt si elles ont entre 18 et 35 ans

les autres ne paient pas d'impôt

Le programme demandera donc l'âge et le sexe de chaque habitant, et se prononcera donc ensuite sur le fait que l'habitant est imposable.

```
#!/bin/bash
echo "donner l'age"
read age
echo "donner le sex 1 pour homme et 0 pour femme"
read sex
if [ $age -ge 20 ] && [ $sex -eq 1 ]
then
echo "homme est imposable"
elif [ $age -ge 18 ] && [ $age -le 35 ] && [ $sex -eq 0 ]
then
echo "femme est imposable"
else
echo "ne paient pas l'impot"
fi
```

```
donner l'age
25
donner le sex 1 pour homme et 0 pour femme
1
homme est imposable
[fz@fedora ~]$ ./script.sh
donner l'age
20
donner le sex 1 pour homme et 0 pour femme
0
femme est imposable
[fz@fedora ~]$ ./script.sh
donner l'age
40
donner le sex 1 pour homme et 0 pour femme
1
homme est imposable
[fz@fedora ~]$ ./script.sh
donner l'age
40
donner le sex 1 pour homme et 0 pour femme
0
ne paient pas l'impot
```

## Exercice 4 :

Écrire un programme qui lit les paramètres d'une équation de premier degré  $ax+b=0$  et affiche la solution

```
#!/bin/bash
echo "donner la valeur de a"
read a
echo "donner la valeur de b"
read b
if [ $a -ne 0 ] && [ $b -ne 0 ]
then
x=$(( (-1*$b) / $a ))
echo "la solution de l'equation est :$x"
elif [ $a -eq 0 ] && [ $b -ne 0 ]
then
echo "n'a pas de solution"
elif [ $a -ne 0 ] && [ $b -eq 0 ]
then
echo " une infinité"
fi
```

### Exercice 5:

A-Écrivez un programme qui calcule la somme des n premiers nombres entiers positifs. Le programme demandera à l'utilisateur d'entrer la valeur de n.

b- Écrivez un programme qui calcule la somme des n premiers nombres entiers positifs paires. Le programme demandera à l'utilisateur d'entrer la valeur de n.

a-

```
#!/bin/bash
echo "donner un nombre entier n"
read n
if [ $n -gt 0 ]
then
somme=$(( $n * ($n - 1)/2))
echo "la somme des n premier nombre positif est : $somme"
elif [ $n -eq 0 ]
then
somme=0
echo "la somme des n premier nombre positif est : $somme"
fi
```

```
[fz@fedora ~]$ ./script.sh
donner un nombre
10
la somme des n premier nombre positifs est : 45
```

b)

```
#!/bin/bash
echo "donner un nombre entier n"
read n
if [ $n -ne 0 ]
then
somme=$(( $n * ($n - 1)))
echo "la somme des n premier nombre positif est : $somme"
else
somme=0
fi
```

```
donner un nombre
10
la somme des n premier nombre positifs pairs est : 90
```

## Les fonctions :

### Exercice 6:

écrire une fonction qui permet de calculer la factorielle d'un nombre .

```
#!/bin/bash
factoriel()
{
  echo "donner un nombre"
  read a
  i=1
  f=1
  while [ $i -le $a ]
  do
    f=$(( $f * $i ))
    i=$(( $i + 1 ))
  done
  echo "le factoriel de $a est :$f"
}
factoriel $a
```

```
donner un nombre
6
le factoriel de 6 est :720
[fz@fedora ~]$
```

### Exercice 7 :

Écrire un programme qui lit un entier et vérifie si cet entier est divisible par 3

```
#!/bin/bash
divisible()
{
  echo "donner un nombre"
  read a
  if [ $(( $a % 3 )) -eq 0 ]
  then
    echo "$a est divisible par 3"
  else
    echo "$a n'est pas divisible par 3"
  fi
}
divisible|
```

```
[fz@fedora ~]$ ./script.sh
donner un nombre
6
6 est divisible par 3
[fz@fedora ~]$
```

### Exercice 8 :

Écrire le programme qui affiche l'une des mentions suivantes :

entre 13 et 16 Bien

entre 0 et 9 insuffisant

entre 10 et 12 passable

entre 17 et 20 Très Bien



```
#!/bin/bash
mentions()
{
echo "donner une note"
read a
if [ $a -ge 0 -a $a -lt 9 ]
then
echo "insuffisant"
elif [ $a -ge 13 -a $a -lt 16 ]
then
echo "bien"
elif [ $a -ge 10 -a $a -lt 12 ]
then
echo "passable"
elif [ $a -ge 17 -a $a -lt 20 ]
then
echo "trés bien"
fi
}
mentions
```

```
[fz@fedora ~]$ ./script.sh
donner une note
15
bien
[fz@fedora ~]$ ./script.sh
donner une note
18
trés bien
[fz@fedora ~]$ ./script.sh
donner une note
5
insuffisant
[fz@fedora ~]$
```

## Exercice 9:

Écrire un programme qui lit trois nombres et dit s'ils sont dans l'ordre croissant.

```
#!/bin/bash
ordre()
{
  echo "donner les 3 nombre"
  read a
  read b
  read c
  if [ $a -gt $b -a $b -gt $c ]
  then
    echo "ordre décroissant"
  else
    echo "ordre croissant"
  fi
}
ordre
```

```
donner les 3 nombre
1
2
3
ordre croissant
[fz@fedora ~]$ ./script.sh
donner les 3 nombre
3
2
1
ordre décroissant
```

### Exercice 10:

Ecrire un programme qui demande un nombre de départ, et qui ensuite affiche les dix nombres suivants. Par exemple, si l'utilisateur entre le nombre 17, le programme affichera les nombres de 18 à 27.

```
#!/bin/bash
dixnombre()
{
  echo "entrer un nombre"
  read n
  echo "les dix nombres après $n est :"
  for i in $(seq $n $((n+9)))
  do
    s=$((i+1))
    echo "$s"
  done
}|
dixnombre
```

```
[fz@fedora ~]$ ./script.sh
entrer un nombre
12
les dix nombres après 12 est :
13
14
15
16
17
18
19
20
21
22
[fz@fedora ~]$
```