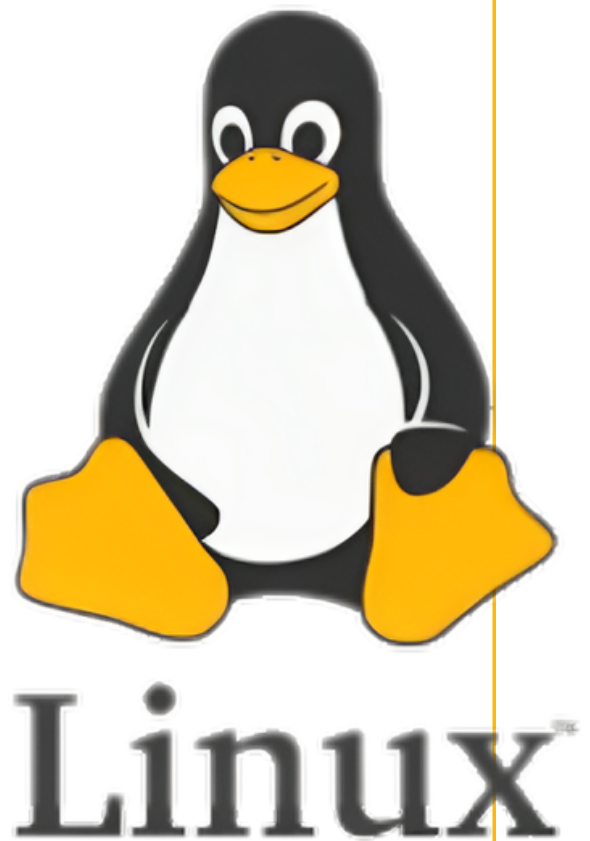


INFORMATIQUE S5

**COMPTE
RENDUE OS
LINUX**

- **SCRIPT
SHELL**



Encadré par MR :

M.MOUKHAF

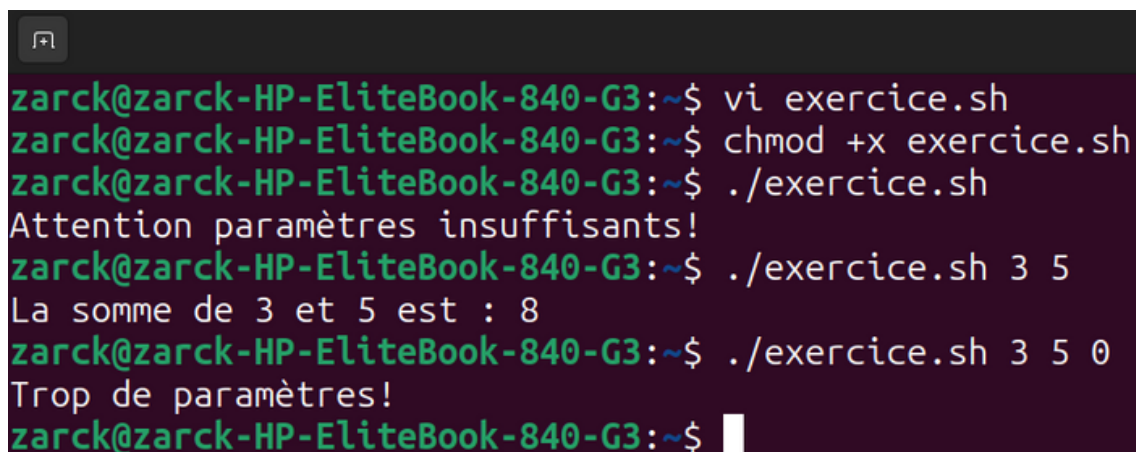
Réalisé par :

El Omari Zakaria

EXERCICE 1 :

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

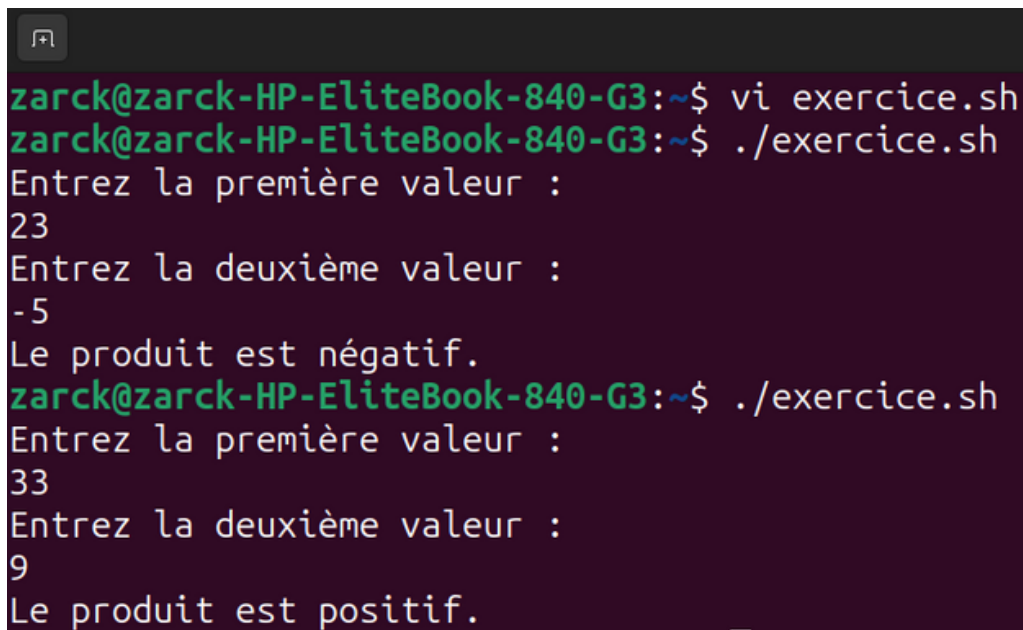
```
# Vérifier le nombre de paramètres
if [ "$#" -lt 2 ]; then
    echo "Attention paramètres insuffisants!"
elif [ "$#" -gt 2 ]; then
    echo "Trop de paramètres!"
else
    # Calculer et afficher la somme
    sum=$(( $1 + $2 ))
    echo "La somme de $1 et $2 est : $sum"
fi
```

A terminal window with a dark background and green text. It shows the user 'zarck' at a machine named 'zarck-HP-EliteBook-840-G3' running several commands. First, 'vi exercice.sh' opens a file. Then, 'chmod +x exercice.sh' makes it executable. Next, './exercice.sh' is run without arguments, resulting in the output 'Attention paramètres insuffisants!'. Then, './exercice.sh 3 5' is run with two arguments, resulting in the output 'La somme de 3 et 5 est : 8'. Finally, './exercice.sh 3 5 0' is run with three arguments, resulting in the output 'Trop de paramètres!'. The prompt returns to the user after each command.

```
zarck@zarck-HP-EliteBook-840-G3:~$ vi exercice.sh
zarck@zarck-HP-EliteBook-840-G3:~$ chmod +x exercice.sh
zarck@zarck-HP-EliteBook-840-G3:~$ ./exercice.sh
Attention paramètres insuffisants!
zarck@zarck-HP-EliteBook-840-G3:~$ ./exercice.sh 3 5
La somme de 3 et 5 est : 8
zarck@zarck-HP-EliteBook-840-G3:~$ ./exercice.sh 3 5 0
Trop de paramètres!
zarck@zarck-HP-EliteBook-840-G3:~$
```

EXERCICE 2 -a :

```
#!/bin/bash
echo "Entrez la première valeur : "
read val1
echo "Entrez la deuxième valeur : "
read val2
# Vérifier si le produit est positif ou négatif
if [ $val1 -gt 0 -a $val2 -gt 0 ] || [ $val1 -lt 0 -a $val2 -lt 0 ]; then
    echo "Le produit est positif."
else
    echo "Le produit est négatif."
fi
```

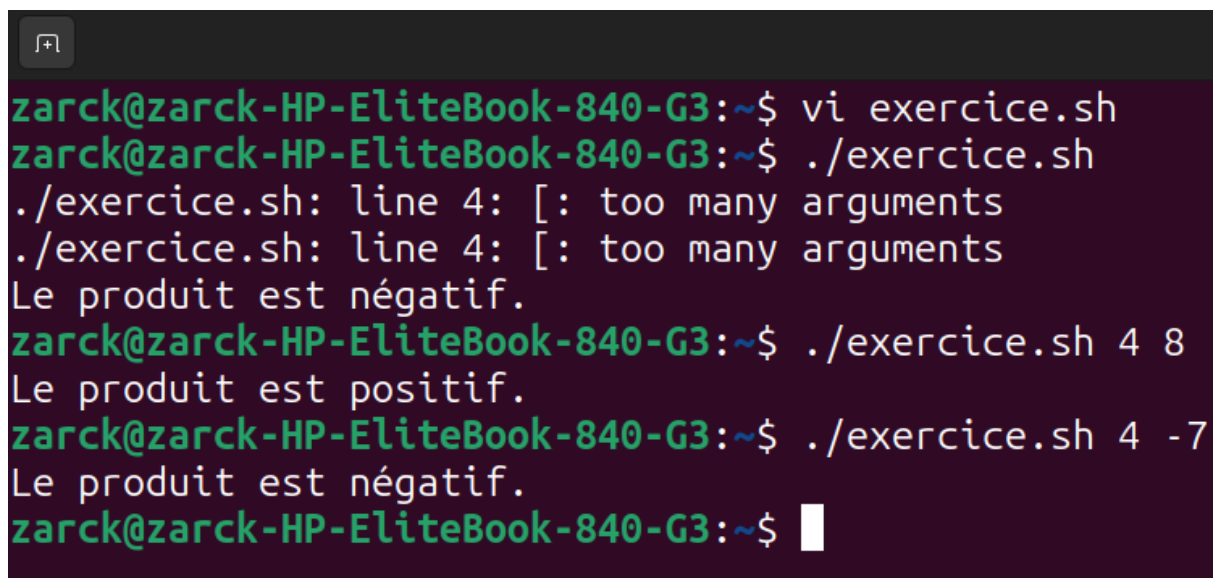


```
zarck@zarck-HP-EliteBook-840-G3:~$ vi exercice.sh
zarck@zarck-HP-EliteBook-840-G3:~$ ./exercice.sh
Entrez la première valeur :
23
Entrez la deuxième valeur :
-5
Le produit est négatif.
zarck@zarck-HP-EliteBook-840-G3:~$ ./exercice.sh
Entrez la première valeur :
33
Entrez la deuxième valeur :
9
Le produit est positif.
```

EXERCICE 2 -b :

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

```
# Vérifier si le produit est positif ou négatif
if [ $1 -gt 0 -a $2 -gt 0 ] || [ $1 -lt 0 -a $2 -lt 0 ]; then
    echo "Le produit est positif."
else
    echo "Le produit est négatif."
fi
```

A terminal window with a dark background and green text. It shows the user editing a file, running it without arguments, and then with two different pairs of arguments to test the logic.

```
zarck@zarck-HP-EliteBook-840-G3:~$ vi exercice.sh
zarck@zarck-HP-EliteBook-840-G3:~$ ./exercice.sh
./exercice.sh: line 4: [: too many arguments
./exercice.sh: line 4: [: too many arguments
Le produit est négatif.
zarck@zarck-HP-EliteBook-840-G3:~$ ./exercice.sh 4 8
Le produit est positif.
zarck@zarck-HP-EliteBook-840-G3:~$ ./exercice.sh 4 -7
Le produit est négatif.
zarck@zarck-HP-EliteBook-840-G3:~$
```

EXERCICE SUR (EXPRESSION ARITHMÉTIQUES)

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

```
a=6
```

```
b=21
```

```
c=9
```

```
sum=$(( $a + $b ))
```

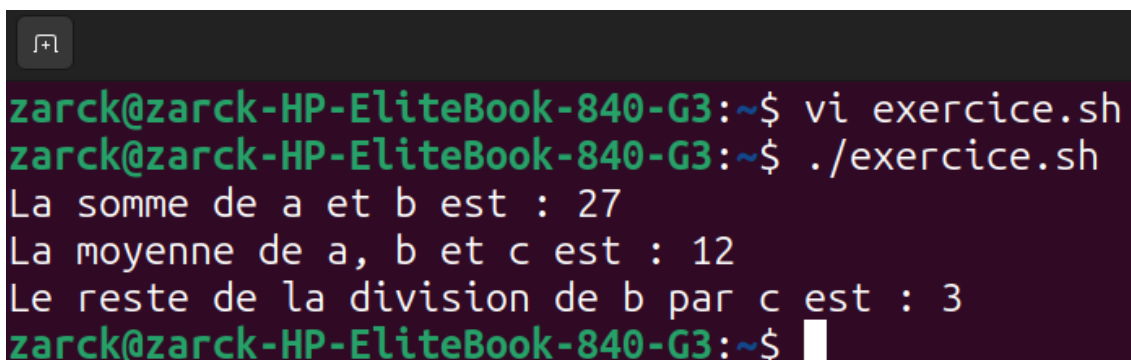
```
echo "La somme de a et b est : $sum"
```

```
moyenne=$(( ($a + $b + $c) / 3 ))
```

```
echo "La moyenne de a, b et c est : $moyenne"
```

```
reste=$(( $b % $c ))
```

```
echo "Le reste de la division de b par c est : $reste"
```

A terminal window with a dark background and light green text. The prompt is 'zarck@zarck-HP-EliteBook-840-G3:~\$'. The user enters 'vi exercice.sh' and then './exercice.sh'. The script outputs three lines: 'La somme de a et b est : 27', 'La moyenne de a, b et c est : 12', and 'Le reste de la division de b par c est : 3'. The prompt returns to 'zarck@zarck-HP-EliteBook-840-G3:~\$' with a cursor.

```
zarck@zarck-HP-EliteBook-840-G3:~$ vi exercice.sh
zarck@zarck-HP-EliteBook-840-G3:~$ ./exercice.sh
La somme de a et b est : 27
La moyenne de a, b et c est : 12
Le reste de la division de b par c est : 3
zarck@zarck-HP-EliteBook-840-G3:~$
```

EXERCICE 3 (CASE ... ESAC)

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

```
# Utiliser case pour afficher le jour correspondant  
case $1 in
```

```
1) echo "C'est Lundi!" ;;
```

```
2) echo "C'est Mardi!" ;;
```

```
3) echo "C'est Mercredi!" ;;
```

```
4) echo "C'est Jeudi!" ;;
```

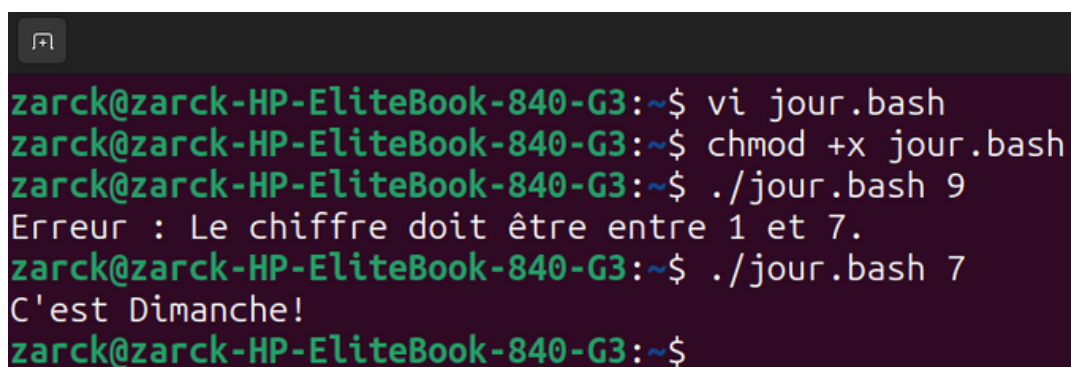
```
5) echo "C'est Vendredi!" ;;
```

```
6) echo "C'est Samedi!" ;;
```

```
7) echo "C'est Dimanche!" ;;
```

```
*) echo "Erreur : Le chiffre doit être entre 1 et 7." ;;
```

```
esac
```



```
zarck@zarck-HP-EliteBook-840-G3:~$ vi jour.bash  
zarck@zarck-HP-EliteBook-840-G3:~$ chmod +x jour.bash  
zarck@zarck-HP-EliteBook-840-G3:~$ ./jour.bash 9  
Erreur : Le chiffre doit être entre 1 et 7.  
zarck@zarck-HP-EliteBook-840-G3:~$ ./jour.bash 7  
C'est Dimanche!  
zarck@zarck-HP-EliteBook-840-G3:~$
```

EXERCICE 4 (WHILE)

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

```
num=0
```

```
while [ "$num" -ne 999 ]; do
```

```
    echo "Entrez un autre nombre (999 pour terminer) : "
```

```
    read num
```

```
    echo "$num" >> numeros.txt
```

```
done
```

```
# Trier les nombres en ordre croissant
```

```
sort -n numeros.txt
```

A terminal window with a dark background and green text. The prompt is 'zarck@zarck-HP-EliteBook-840-G3:~\$'. The user runs './exercice.sh'. The script prompts 'Entrez un autre nombre (999 pour terminer) :'. The user enters '232', '44', '86', and '999' in sequence. The script then displays the sorted numbers: '44', '86', '232', and '999'. The prompt returns to 'zarck@zarck-HP-EliteBook-840-G3:~\$' with a cursor.

```
zarck@zarck-HP-EliteBook-840-G3:~$ ./exercice.sh
Entrez un autre nombre (999 pour terminer) :
232
Entrez un autre nombre (999 pour terminer) :
44
Entrez un autre nombre (999 pour terminer) :
86
Entrez un autre nombre (999 pour terminer) :
999
44
86
232
999
zarck@zarck-HP-EliteBook-840-G3:~$
```

EXERCICE 5 (FOR)

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

```
# Vérifier le nombre de paramètres
```

```
if [ "$#" -ne 1 ]; then
```

```
    echo "Erreur : Veuillez fournir la chaîne de caractères en  
    tant que paramètre."
```

```
    exit 1
```

```
fi
```

```
files=$(find . -name "*.java" -exec grep -l "$1" {} +)
```

```
# Afficher les fichiers
```

```
echo "Fichiers Java contenant la chaîne '$1' :"
```

```
echo "$files"
```



```
zarck@zarck-HP-EliteBook-840-G3:~$ vi exercice.sh
zarck@zarck-HP-EliteBook-840-G3:~$ ./exercice.sh
Erreur : Veuillez fournir la chaîne de caractères en tant que paramètre.
zarck@zarck-HP-EliteBook-840-G3:~$ ./exercice.sh MaClasse
Fichiers Java contenant la chaîne 'MaClasse' :
./eclipse-workspace/threads/src/Presentation/MaClasse.java
./eclipse-workspace/threads/src/Presentation/MaClasse2.java
./eclipse-workspace/TP4/src/Presentation/MaClasse.java
./eclipse-workspace/TP4/src/Presentation/MaClasse2.java
zarck@zarck-HP-EliteBook-840-G3:~$
```


EXERCICE 6 (UNTIL)

```
#!/bin/bash
# Vérifier le nombre d'arguments
if [ "$#" -ne 1 ]; then
    echo "Erreur : Veuillez fournir un seul argument."
    exit 1
fi

# Inverser le nombre
num=$1
inverse=""

until [ "$num" -eq 0 ]; do
    d=$(( $num % 10 ))
    inverse="${inverse}${d}"
    num=$(( $num / 10 ))
done
echo "L'inverse du nombre $num est : $inverse"
```

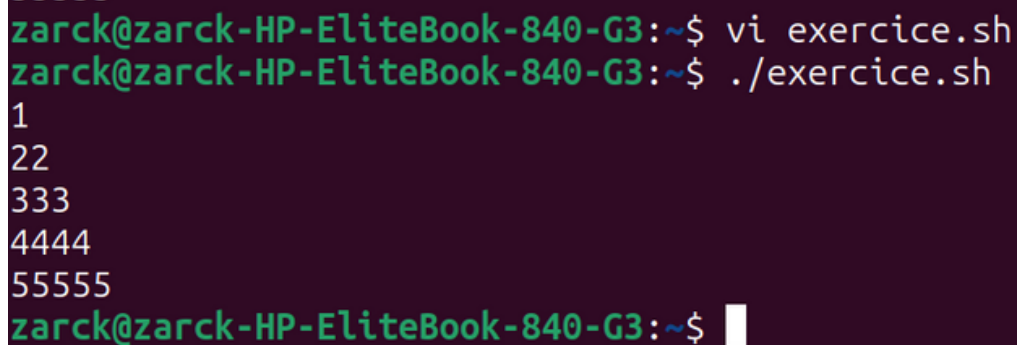
```
zarck@zarck-HP-EliteBook-840-G3:~$ vi exercice.sh
zarck@zarck-HP-EliteBook-840-G3:~$ ./exercice.sh 7543
L'inverse du nombre 0 est : 3457
zarck@zarck-HP-EliteBook-840-G3:~$ █
```

EXERCICE 7 (WHILE)

```
#!/bin/bash
nbr_lignes=5
ligne=1

while [ $ligne -le $nbr_lignes ]; do
    i=1
    while [ $i -le $ligne ]; do
        echo -n "$ligne"
        i=$((i + 1))
    done

    echo
    ligne=$((ligne + 1))
done
```

A terminal window with a dark purple background. The prompt is 'zarck@zarck-HP-EliteBook-840-G3:~\$'. The user enters 'vi exercice.sh' and then './exercice.sh'. The output shows five lines: '1', '22', '333', '4444', and '55555'. The prompt returns to 'zarck@zarck-HP-EliteBook-840-G3:~\$' with a cursor.

```
zarck@zarck-HP-EliteBook-840-G3:~$ vi exercice.sh
zarck@zarck-HP-EliteBook-840-G3:~$ ./exercice.sh
1
22
333
4444
55555
zarck@zarck-HP-EliteBook-840-G3:~$
```

EXERCICE PROBLÈME

Un script qui demande à l'utilisateur d'entrer un nom de fichier, recherche tous les fichiers correspondant au nom dans le répertoire, puis calcule et affiche la somme des tailles de ces fichiers. Si le répertoire n'existe pas, le script affiche un message d'erreur.

```
zarck@zarck-HP-EliteBook-840-G3:~$ ./exercice.sh
Entrez le chemin du répertoire :
/home/zarck/Downloads
Entrez le nom du fichier ou l'extension :
*.pdf
/home/zarck/Downloads/TP_3.1.pdf
/home/zarck/Downloads/flinfo_S5_SE_CR3_elomari_zakaria-1.pdf
/home/zarck/Downloads/flinfo_S5_P00_CR4_elomari_zakaria.pdf
/home/zarck/Downloads/flinfo_SE_CR3_zakaria_elomari.pdf
/home/zarck/Downloads/flinfo_S5_P00_CR5_elomari_zakaria.pdf
/home/zarck/Downloads/TP_4.1.pdf
/home/zarck/Downloads/Cours_SE_linux_final.pdf
/home/zarck/Downloads/Rapport_1.1.pdf-2_pagenumbr.pdf
La somme des tailles des fichiers correspondant au nom '*.pdf' dans le répertoire '/home/zarck/Downloads' est : 18492836 octets.
zarck@zarck-HP-EliteBook-840-G3:~$
```

EXERCICE PROBLÈME

```
#!/bin/bash
# Demander à l'utilisateur de saisir un répertoire
echo "Entrez le chemin du répertoire : "
read directory

# Vérifier si le répertoire existe
if [ ! -d "$directory" ]; then
    echo "Erreur : Le répertoire n'existe pas."
    exit 1
fi

# Initialiser la somme
sum=0

# Demander à l'utilisateur de saisir le nom de fichier
echo "Entrez un motif de fichier : "
read nom

# Utiliser find pour trouver les fichiers correspondant au nom dans le
répertoire
fichier=$(find "$directory" -name "$nom")

# Vérifier si des fichiers ont été trouvés
if [ -z "$fichier" ]; then
    echo "Aucun fichier trouvé pour le nom '$nom' dans le répertoire
'$directory'."
else
    # Calculer la somme des tailles des fichiers
    for fichier in $fichier; do
        echo "$fichier"
        taille=$(wc -c < "$fichier")
        sum=$((sum + taille))
    done

    # Afficher la somme des tailles des fichiers
    echo "La somme des tailles des fichiers correspondant au nom '$nom' dans
le répertoire '$directory' est : $sum octets."
fi
```