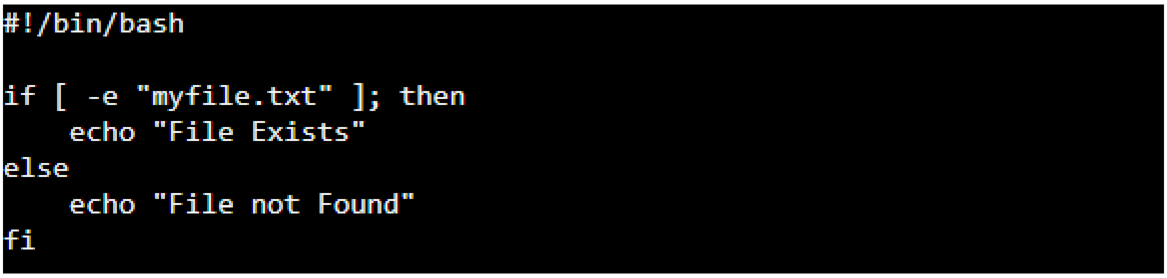
Day 7 Assignment

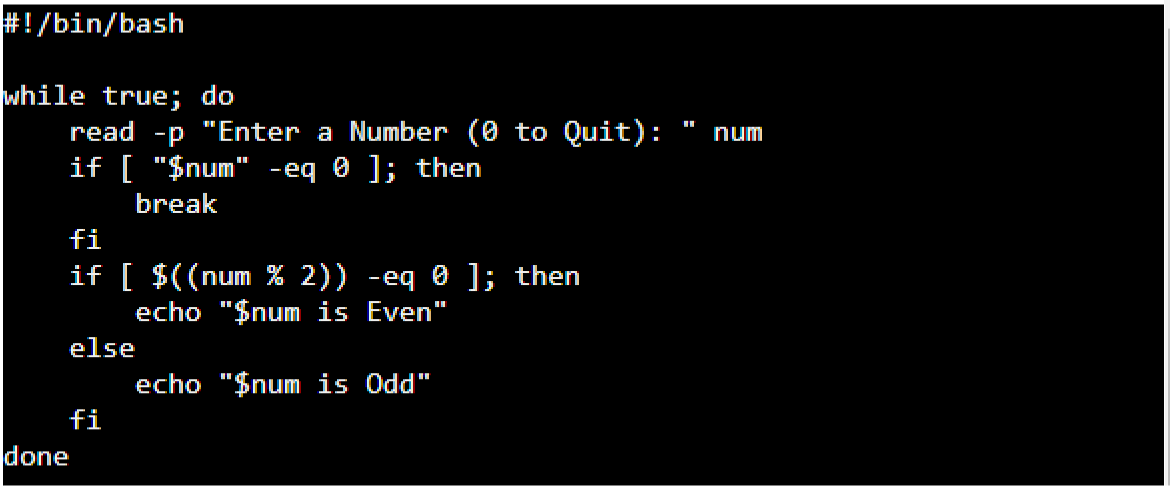
Name : Mehul Anjikhane   Email: mehulanjikhane13@gamil.com

Assignment 1: Ensure the script checks if a specific file (e.g., myfile.txt) exists in the current directory. If it exists, print "File exists", otherwise print "File not found".

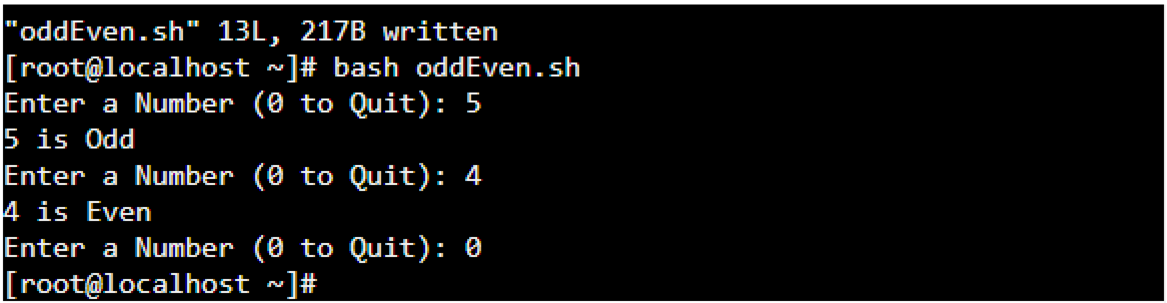


```
#!/bin/bash

if [ -e "myfile.txt" ]; then
    echo "File Exists"
else
    echo "File not Found"
fi
```

Assignment 2: Write a script that reads numbers from the user until they enter '0'. The script should also print whether each number is odd or even.

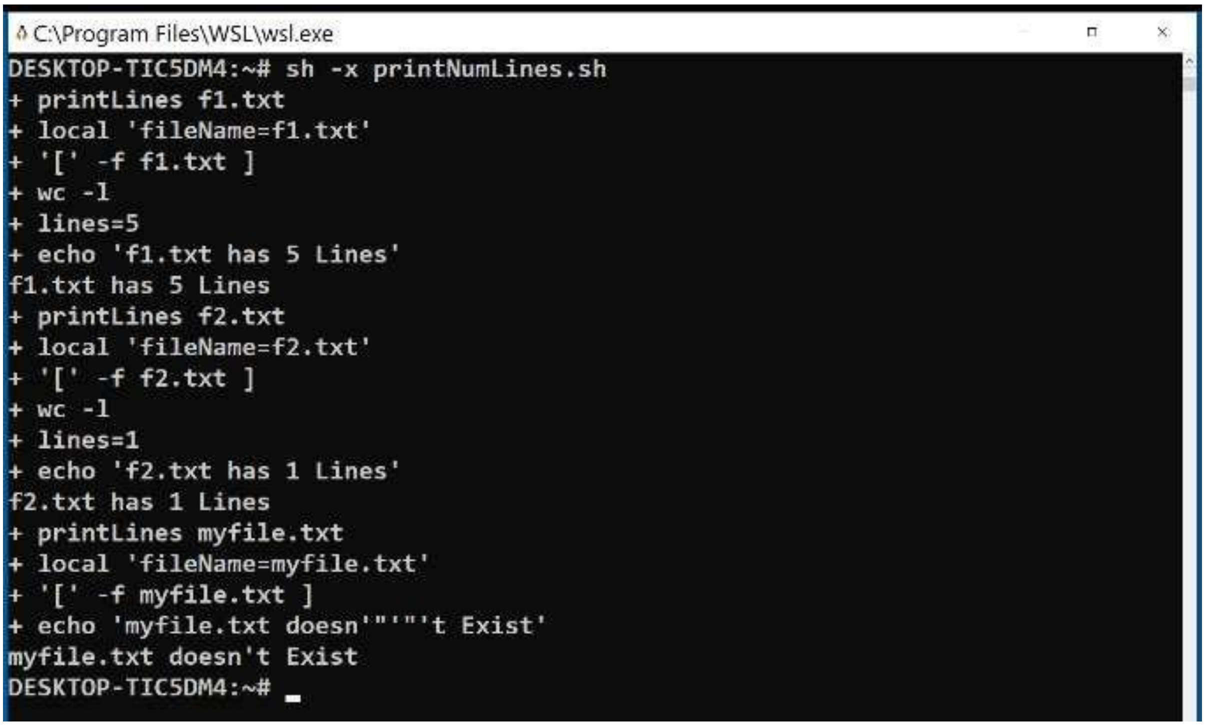


```
#!/bin/bash

while true; do
    read -p "Enter a Number (0 to Quit): " num
    if [ "$num" -eq 0 ]; then
        break
    fi
    if [ $((num % 2)) -eq 0 ]; then
        echo "$num is Even"
    else
        echo "$num is Odd"
    fi
done
```

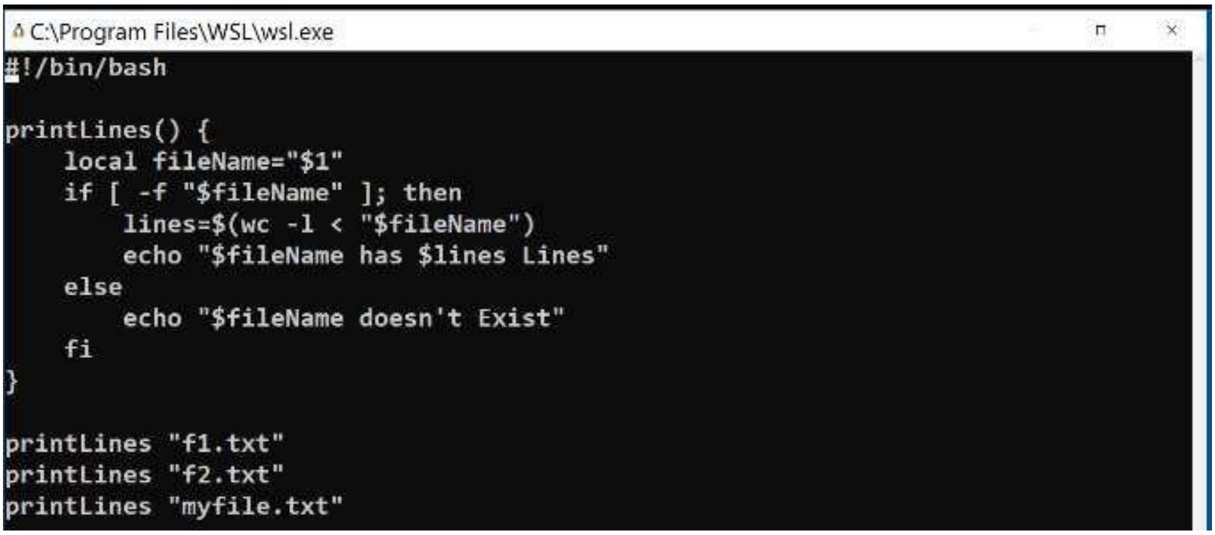


```
"oddEven.sh" 13L, 217B written
[root@localhost ~]# bash oddEven.sh
Enter a Number (0 to Quit): 5
5 is Odd
Enter a Number (0 to Quit): 4
4 is Even
Enter a Number (0 to Quit): 0
[root@localhost ~]#
```

Assignment 3: Create a function that takes a filename as an argument and prints the number of lines in the file. Call this function from your script with different filenames.

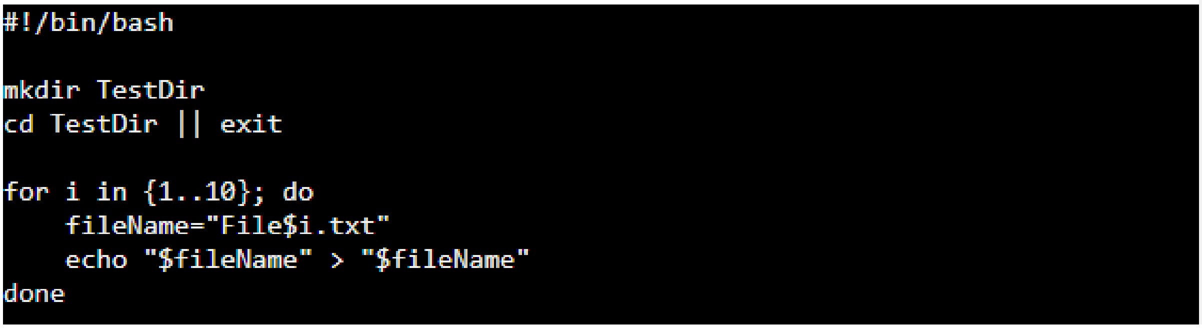


```
DESKTOP-TIC5DM4:~# sh -x printNumLines.sh
+ printLines f1.txt
+ local 'fileName=f1.txt'
+ '[' -f f1.txt ]
+ wc -l
+ lines=5
+ echo 'f1.txt has 5 Lines'
f1.txt has 5 Lines
+ printLines f2.txt
+ local 'fileName=f2.txt'
+ '[' -f f2.txt ]
+ wc -l
+ lines=1
+ echo 'f2.txt has 1 Lines'
f2.txt has 1 Lines
+ printLines myfile.txt
+ local 'fileName=myfile.txt'
+ '[' -f myfile.txt ]
+ echo 'myfile.txt doesn'"'"'t Exist'
myfile.txt doesn't Exist
DESKTOP-TIC5DM4:~# _
```

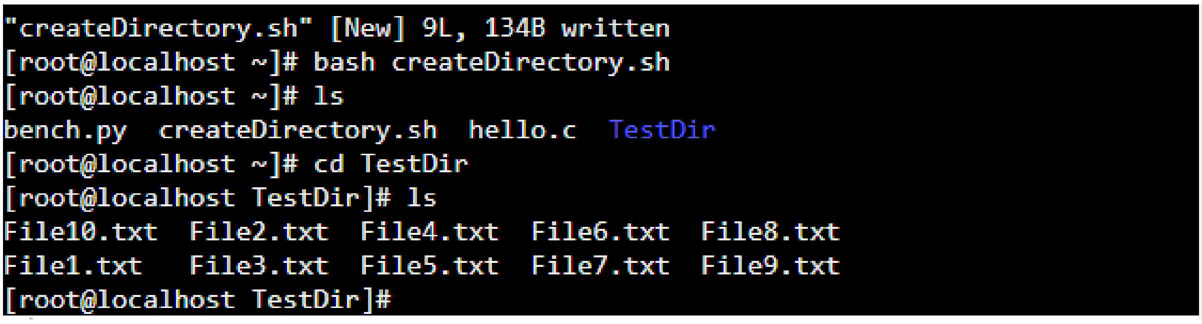


```
#!/bin/bash

printLines() {
    local fileName="$1"
    if [ -f "$fileName" ]; then
        lines=$(wc -l < "$fileName")
        echo "$fileName has $lines Lines"
    else
        echo "$fileName doesn't Exist"
    fi
}

printLines "f1.txt"
printLines "f2.txt"
printLines "myfile.txt"
```

Assignment 4: Write a script that creates a directory named TestDir and inside it, creates ten files named File1.txt, File2.txt, ... File10.txt. Each file should contain its filename as its content (e.g., File1.txt contains "File1.txt")



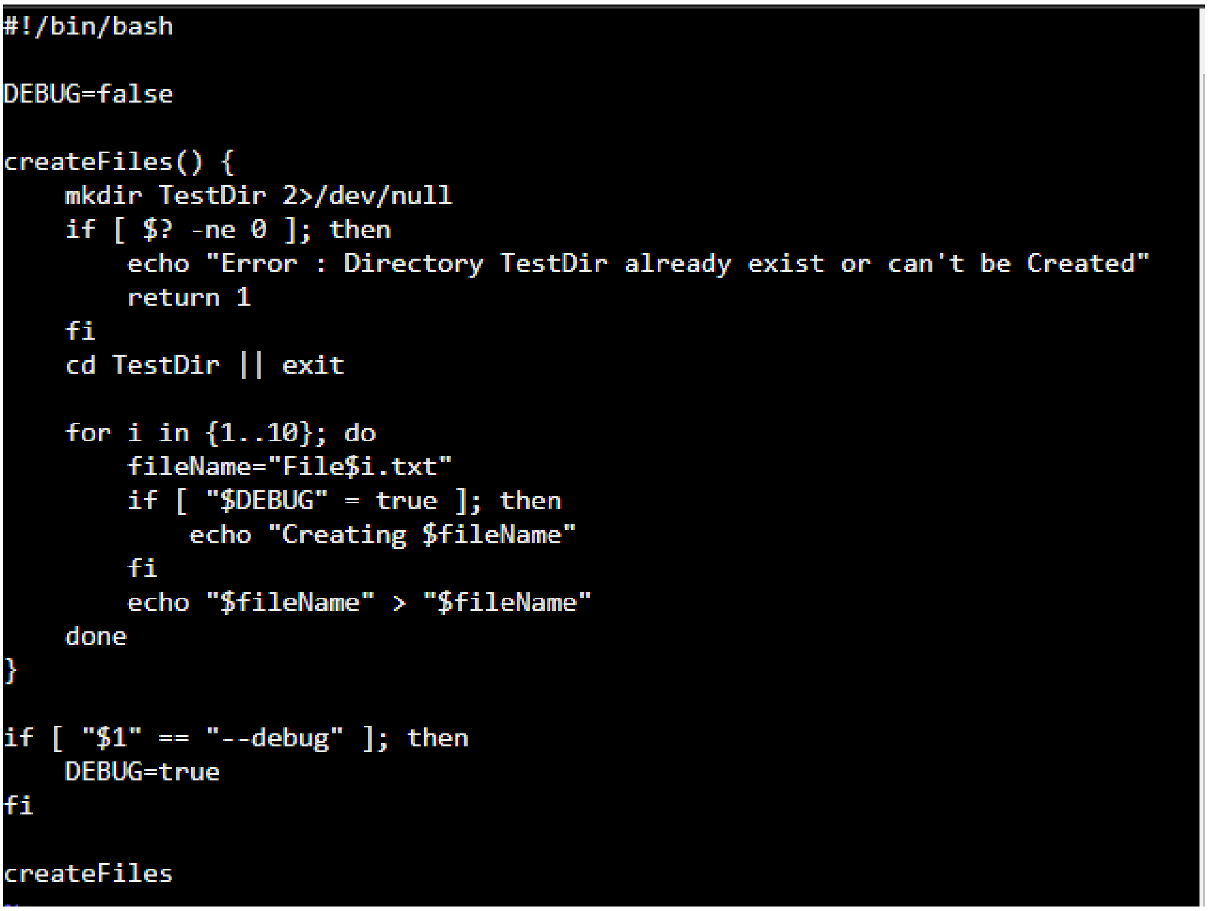
```
#!/bin/bash

mkdir TestDir
cd TestDir || exit

for i in {1..10}; do
    fileName="File$i.txt"
    echo "$fileName" > "$fileName"
done
```



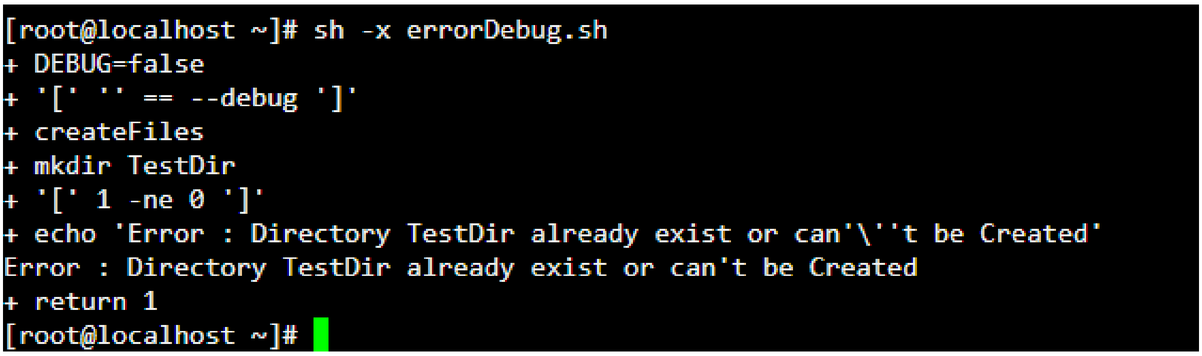
```
"createDirectory.sh" [New] 9L, 134B written
[root@localhost ~]# bash createDirectory.sh
[root@localhost ~]# ls
bench.py  createDirectory.sh  hello.c  TestDir
[root@localhost ~]# cd TestDir
[root@localhost TestDir]# ls
File10.txt  File2.txt  File4.txt  File6.txt  File8.txt
File1.txt   File3.txt  File5.txt  File7.txt  File9.txt
[root@localhost TestDir]#
```

.

Assignment 5: Modify the script to handle errors, such as the directory already existing or lacking permissions to create files. Add a debugging mode that prints additional information when enabled.



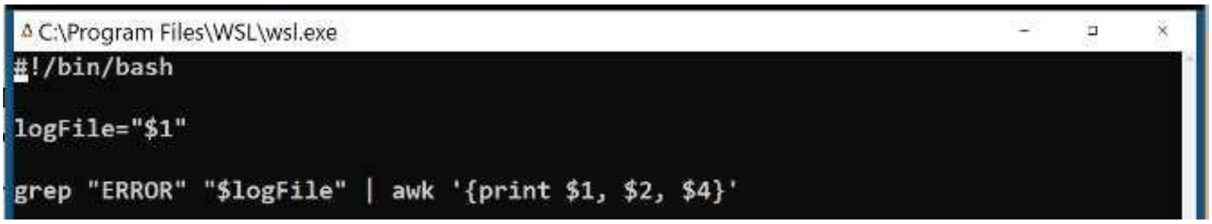
```
#!/bin/bash

DEBUG=false

createFiles() {
    mkdir TestDir 2>/dev/null
    if [ $? -ne 0 ]; then
        echo "Error : Directory TestDir already exist or can't be Created"
        return 1
    fi
    cd TestDir || exit

    for i in {1..10}; do
        fileName="File$i.txt"
        if [ "$DEBUG" = true ]; then
            echo "Creating $fileName"
        fi
        echo "$fileName" > "$fileName"
    done
}

if [ "$1" == "--debug" ]; then
    DEBUG=true
fi

createFiles
```



```
[root@localhost ~]# sh -x errorDebug.sh
+ DEBUG=false
+ '[' '' == --debug ']'
+ createFiles
+ mkdir TestDir
+ '[' 1 -ne 0 ']'
+ echo 'Error : Directory TestDir already exist or can'\''t be Created'
Error : Directory TestDir already exist or can't be Created
+ return 1
[root@localhost ~]#
```

Assignment 6: Given a sample log file, write a script using grep to extract all lines containing "ERROR". Use awk to print the date, time, and error message of each extracted line.

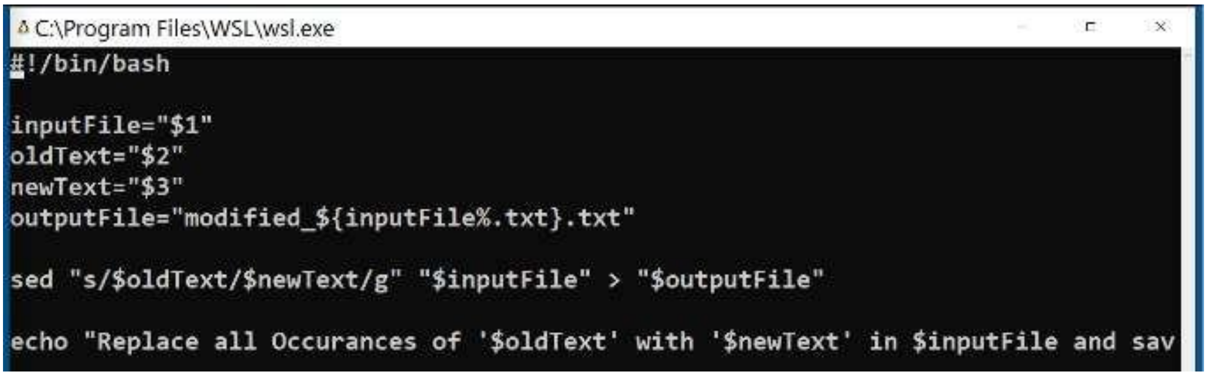
Data Processing with sed



C:\Program Files\WSL\wsl.exe

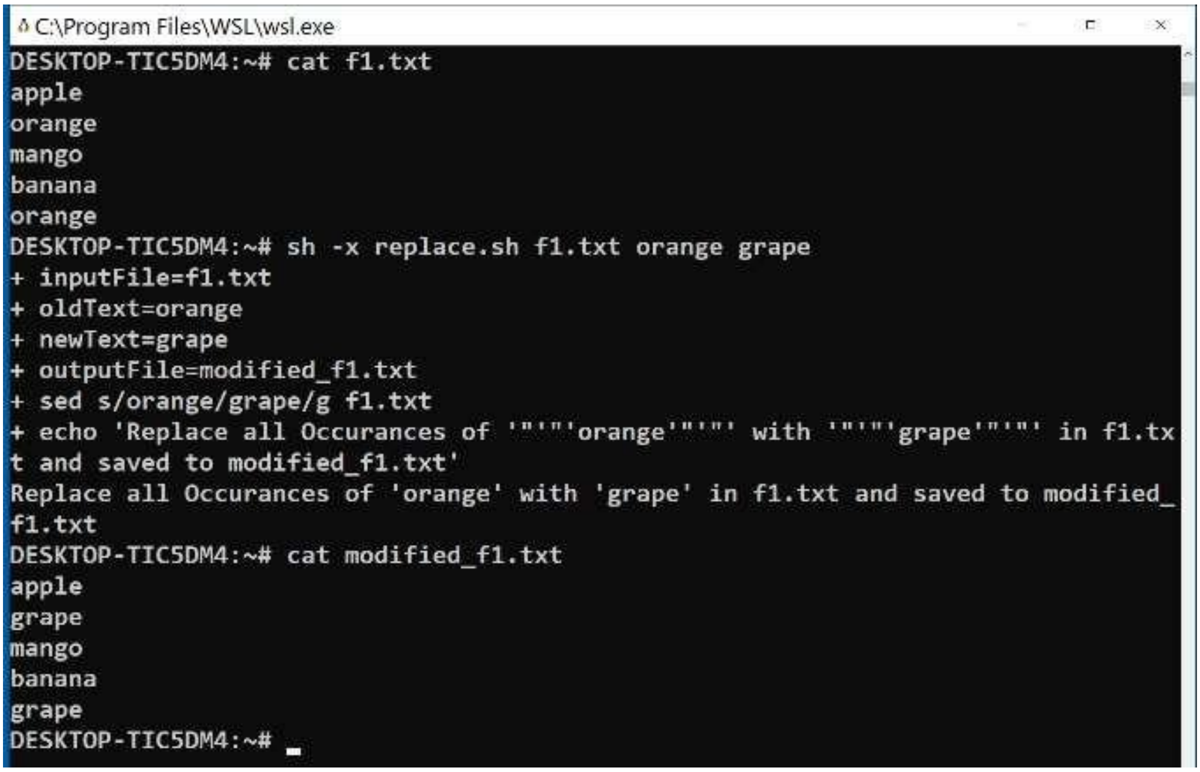
```
#!/bin/bash

logFile="$1"

grep "ERROR" "$logFile" | awk '{print $1, $2, $4}'
```

Assignment 7: Create a script that takes a text file and replaces all occurrences of "old_text" with "new_text". Use sed to perform this operation and output the result to a new file.



C:\Program Files\WSL\wsl.exe

```
#!/bin/bash

inputFile="$1"
oldText="$2"
newText="$3"
outputFile="modified_${inputFile%.txt}.txt"

sed "s/$oldText/$newText/g" "$inputFile" > "$outputFile"

echo "Replace all Occurances of '$oldText' with '$newText' in $inputFile and sav
```



C:\Program Files\WSL\wsl.exe

```
DESKTOP-TIC5DM4:~# cat f1.txt
apple
orange
mango
banana
orange
DESKTOP-TIC5DM4:~# sh -x replace.sh f1.txt orange grape
+ inputFile=f1.txt
+ oldText=orange
+ newText=grape
+ outputFile=modified_f1.txt
+ sed s/orange/grape/g f1.txt
+ echo 'Replace all Occurances of '"'"'orange'"'"' with '"'"'grape'"'"' in f1.tx
t and saved to modified_f1.txt'
Replace all Occurances of 'orange' with 'grape' in f1.txt and saved to modified_
f1.txt
DESKTOP-TIC5DM4:~# cat modified_f1.txt
apple
grape
mango
banana
grape
DESKTOP-TIC5DM4:~# _
```