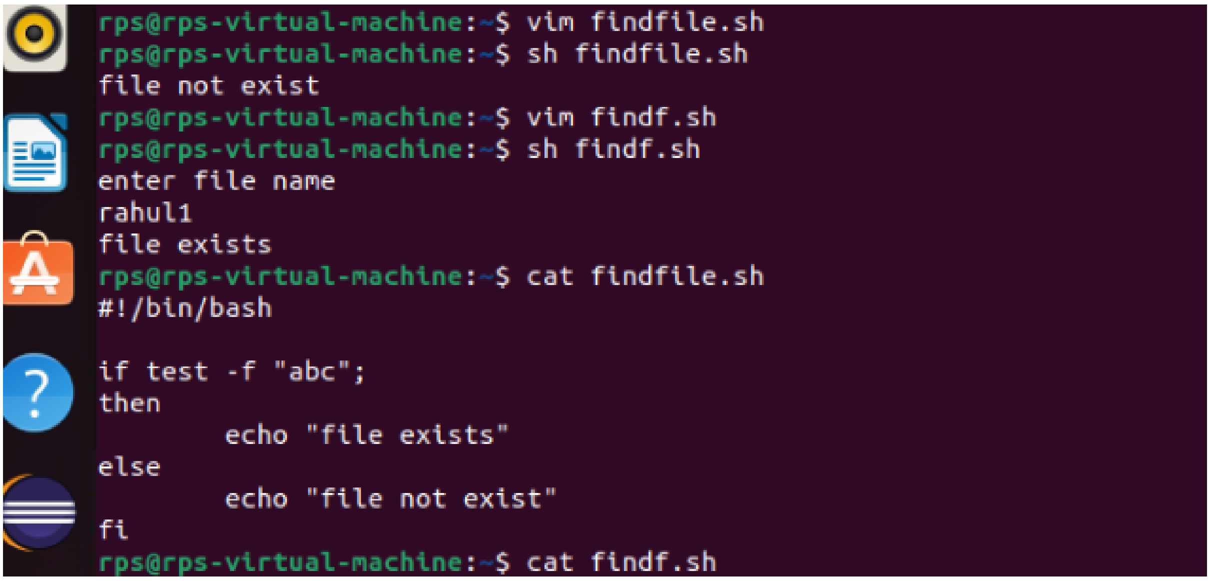
Assignment 6.1

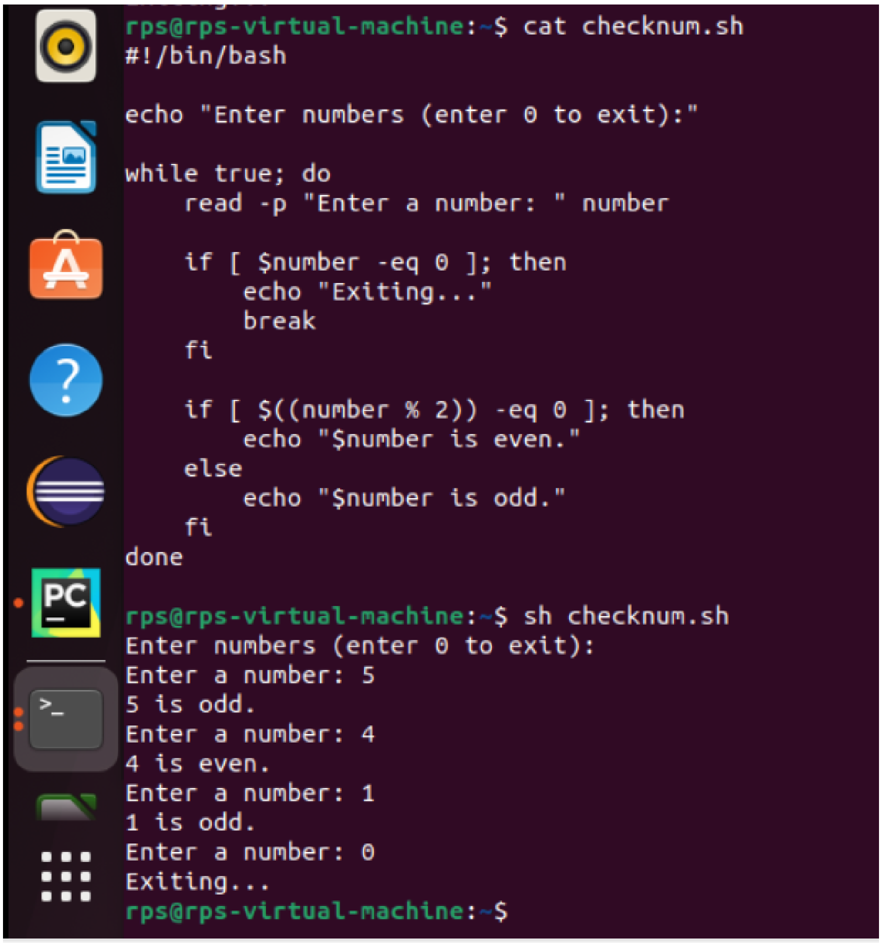
Given: Ensure the script checks if a specific file (e.g., myfile.txt) exists in the current directory. If it exists, print "File exists", otherwise print "File not found".



```
rps@rps-virtual-machine:~$ vim findfile.sh
rps@rps-virtual-machine:~$ sh findfile.sh
file not exist
rps@rps-virtual-machine:~$ vim findf.sh
rps@rps-virtual-machine:~$ sh findf.sh
enter file name
rahul1
file exists
rps@rps-virtual-machine:~$ cat findfile.sh
#!/bin/bash

if test -f "abc";
then
        echo "file exists"
else
        echo "file not exist"
fi
rps@rps-virtual-machine:~$ cat findf.sh
```

Assignment 6.2

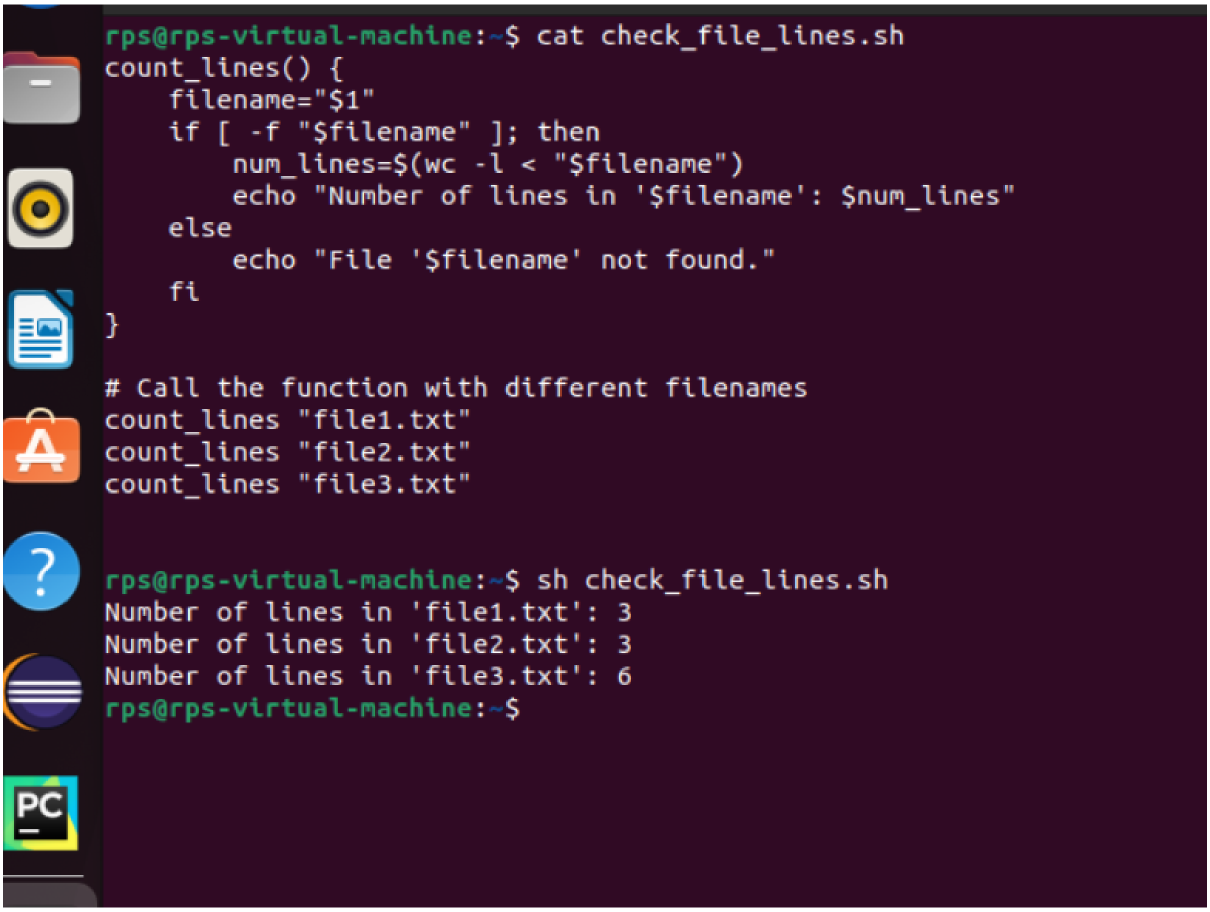
Given: Write a script that reads numbers from the user until they enter '0'. The script should also print whether each number is odd or even.



```
rps@rps-virtual-machine:~$ cat checknum.sh
#!/bin/bash

echo "Enter numbers (enter 0 to exit):"

while true; do
    read -p "Enter a number: " number

    if [ $number -eq 0 ]; then
        echo "Exiting..."
        break
    fi

    if [ $((number % 2)) -eq 0 ]; then
        echo "$number is even."
    else
        echo "$number is odd."
    fi
done

rps@rps-virtual-machine:~$ sh checknum.sh
Enter numbers (enter 0 to exit):
Enter a number: 5
5 is odd.
Enter a number: 4
4 is even.
Enter a number: 1
1 is odd.
Enter a number: 0
Exiting...
rps@rps-virtual-machine:~$
```

Assignment 6.3

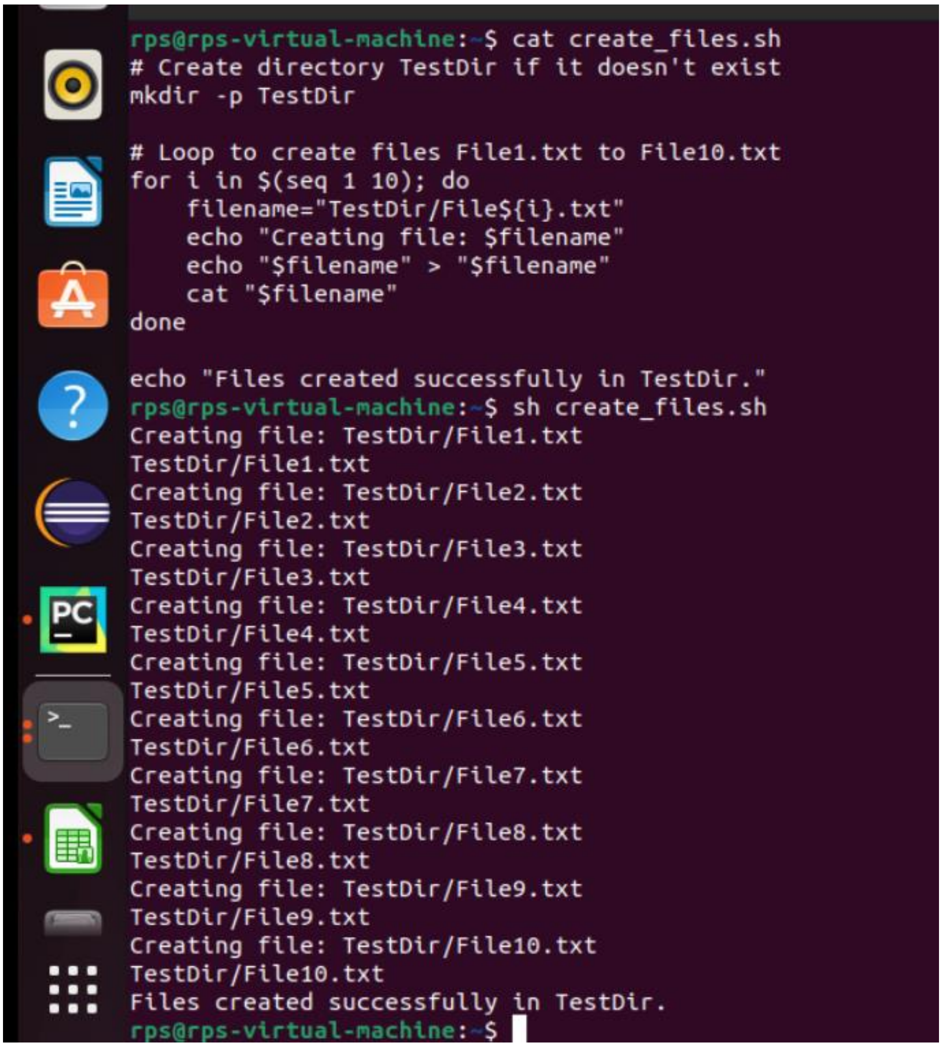
Given: Create a function that takes a filename as an argument and prints the number of lines in the file. Call this function from your script with different filenames.



```
rps@rps-virtual-machine:~$ cat check_file_lines.sh
count_lines() {
    filename="$1"
    if [ -f "$filename" ]; then
        num_lines=$(wc -l < "$filename")
        echo "Number of lines in '$filename': $num_lines"
    else
        echo "File '$filename' not found."
    fi
}

# Call the function with different filenames
count_lines "file1.txt"
count_lines "file2.txt"
count_lines "file3.txt"


rps@rps-virtual-machine:~$ sh check_file_lines.sh
Number of lines in 'file1.txt': 3
Number of lines in 'file2.txt': 3
Number of lines in 'file3.txt': 6
rps@rps-virtual-machine:~$
```

Assignment 6.4

Given: 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").



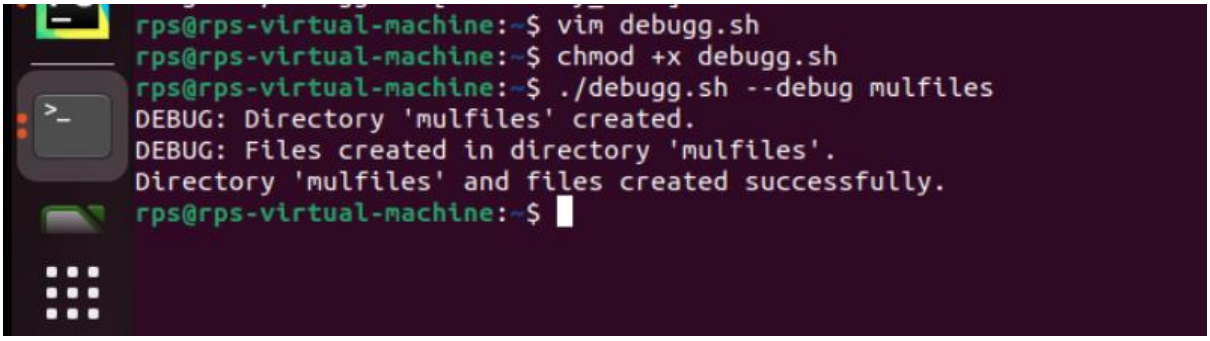
```
rps@rps-virtual-machine:~$ cat create_files.sh
# Create directory TestDir if it doesn't exist
mkdir -p TestDir

# Loop to create files File1.txt to File10.txt
for i in $(seq 1 10); do
    filename="TestDir/File${i}.txt"
    echo "Creating file: $filename"
    echo "$filename" > "$filename"
    cat "$filename"
done

echo "Files created successfully in TestDir."
rps@rps-virtual-machine:~$ sh create_files.sh
Creating file: TestDir/File1.txt
TestDir/File1.txt
Creating file: TestDir/File2.txt
TestDir/File2.txt
Creating file: TestDir/File3.txt
TestDir/File3.txt
Creating file: TestDir/File4.txt
TestDir/File4.txt
Creating file: TestDir/File5.txt
TestDir/File5.txt
Creating file: TestDir/File6.txt
TestDir/File6.txt
Creating file: TestDir/File7.txt
TestDir/File7.txt
Creating file: TestDir/File8.txt
TestDir/File8.txt
Creating file: TestDir/File9.txt
TestDir/File9.txt
Creating file: TestDir/File10.txt
TestDir/File10.txt
Files created successfully in TestDir.
rps@rps-virtual-machine:~$
```

Assignment 6.5

Given: 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
# Function to display usage
display_usage() {
  echo "Usage: $0 [--debug] directory_name"
}

# Function for debugging output
debug() {
  if [ "$DEBUG" = true ]; then
    echo "DEBUG: $1"
  fi
}

# Function to create directory and files
create_directory_and_files() {
  local dir_name=$1

  # Check if directory already exists
  if [ -d "$dir_name" ]; then
    echo "Error: Directory '$dir_name' already exists."
    exit 1
  fi

  # Create directory
  mkdir "$dir_name"
  debug "Directory '$dir_name' created."

  # Check if directory creation successful
  if [ $? -ne 0 ]; then
    echo "Error: Failed to create directory '$dir_name'."
    exit 1
  fi

  # Create files
  touch "$dir_name/file1.txt" "$dir_name/file2.txt" "$dir_name/file3.txt"
  debug "Files created in directory '$dir_name'."

  # Check if file creation successful
  if [ $? -ne 0 ]; then
    echo "Error: Failed to create files in directory '$dir_name'."
    exit 1
  fi

  echo "Directory '$dir_name' and files created successfully."
}

# Main script
if [ $# -lt 1 ]; then
   display_usage
   exit 1
fi

# Check if debugging mode enabled
if [ "$1" = "--debug" ]; then
   DEBUG=true
   shift
fi

# Create directory and files
create_directory_and_files "$1"
```

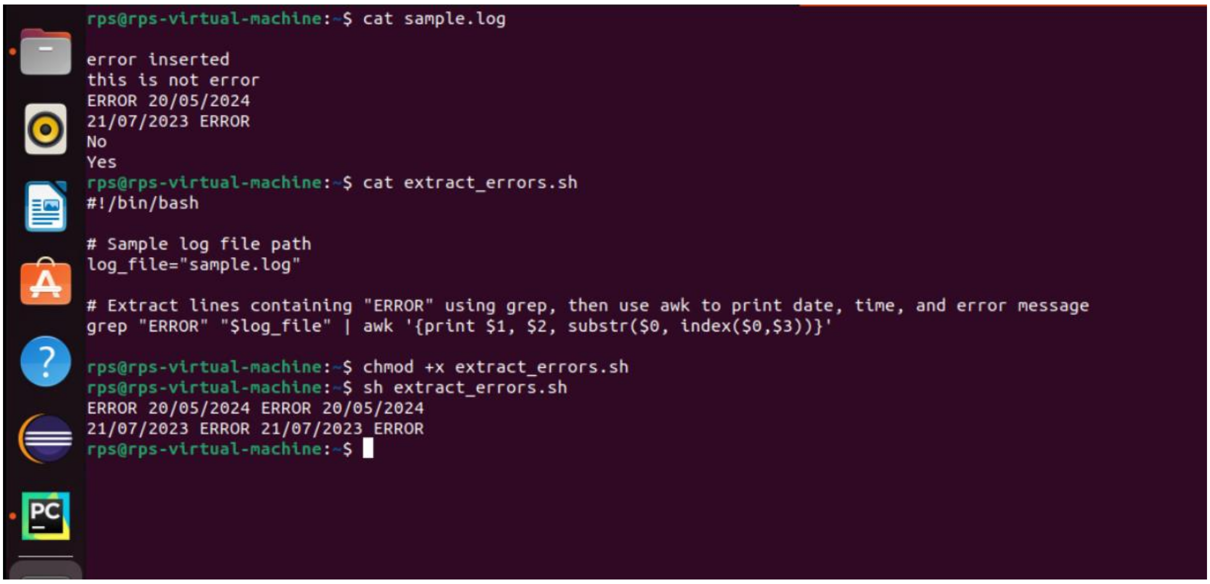
Output:



```
rps@rps-virtual-machine:~$ vim debugg.sh
rps@rps-virtual-machine:~$ chmod +x debugg.sh
rps@rps-virtual-machine:~$ ./debugg.sh --debug mulfiles
DEBUG: Directory 'mulfiles' created.
DEBUG: Files created in directory 'mulfiles'.
Directory 'mulfiles' and files created successfully.
rps@rps-virtual-machine:~$
```

Assignment 6.6

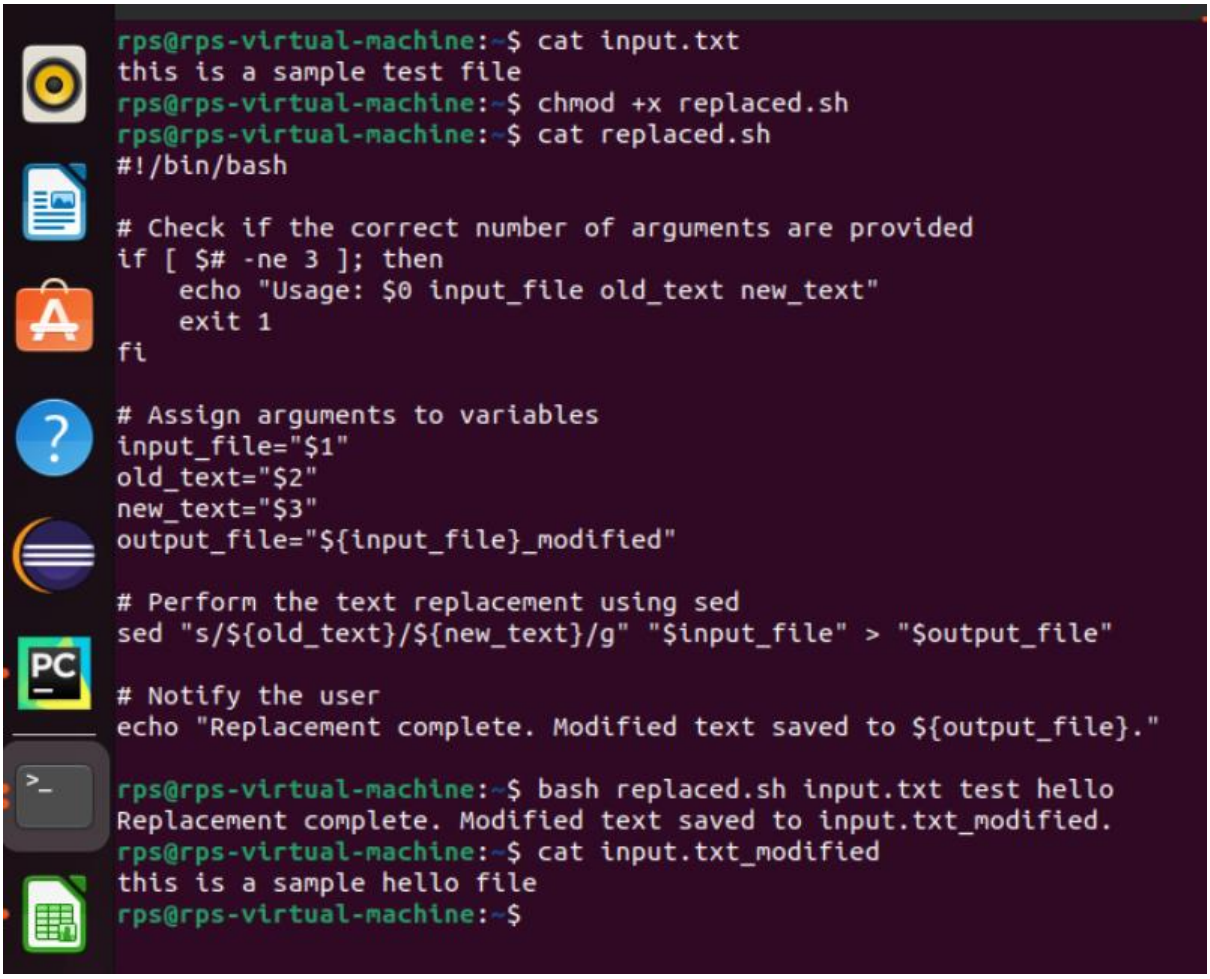
Given: A sample log file, write a script using grep to extract all lines containing "ERROR". Use awk to print the date, time, and error message of each extracted line. Data Processing with sed



```
rps@rps-virtual-machine:~$ cat sample.log

error inserted
this is not error
ERROR 20/05/2024
21/07/2023 ERROR
No
Yes
rps@rps-virtual-machine:~$ cat extract_errors.sh
#!/bin/bash

# Sample log file path
log_file="sample.log"

# Extract lines containing "ERROR" using grep, then use awk to print date, time, and error message
grep "ERROR" "$log_file" | awk '{print $1, $2, substr($0, index($0,$3))}'

rps@rps-virtual-machine:~$ chmod +x extract_errors.sh
rps@rps-virtual-machine:~$ sh extract_errors.sh
ERROR 20/05/2024 ERROR 20/05/2024
21/07/2023 ERROR 21/07/2023 ERROR
rps@rps-virtual-machine:~$ 
```

Assignment 6.7

Given: 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.



```
rps@rps-virtual-machine:~$ cat input.txt
this is a sample test file
rps@rps-virtual-machine:~$ chmod +x replaced.sh
rps@rps-virtual-machine:~$ cat replaced.sh
#!/bin/bash

# Check if the correct number of arguments are provided
if [ $# -ne 3 ]; then
    echo "Usage: $0 input_file old_text new_text"
    exit 1
fi

# Assign arguments to variables
input_file="$1"
old_text="$2"
new_text="$3"
output_file="${input_file}_modified"

# Perform the text replacement using sed
sed "s/${old_text}/${new_text}/g" "$input_file" > "$output_file"

# Notify the user
echo "Replacement complete. Modified text saved to ${output_file}."

rps@rps-virtual-machine:~$ bash replaced.sh input.txt test hello
Replacement complete. Modified text saved to input.txt_modified.
rps@rps-virtual-machine:~$ cat input.txt_modified
this is a sample hello file
rps@rps-virtual-machine:~$
```