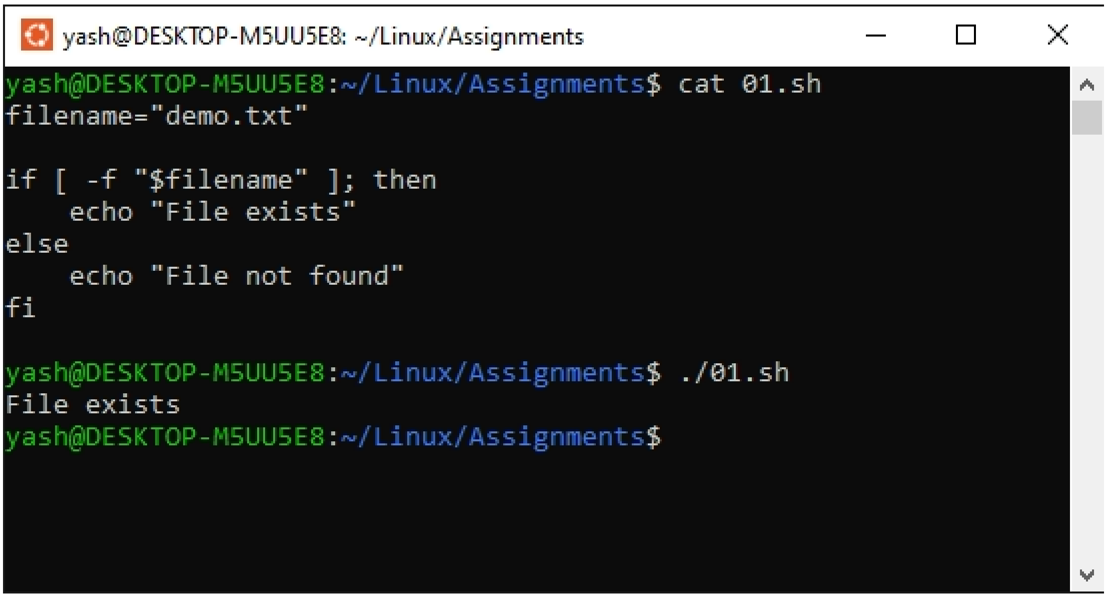
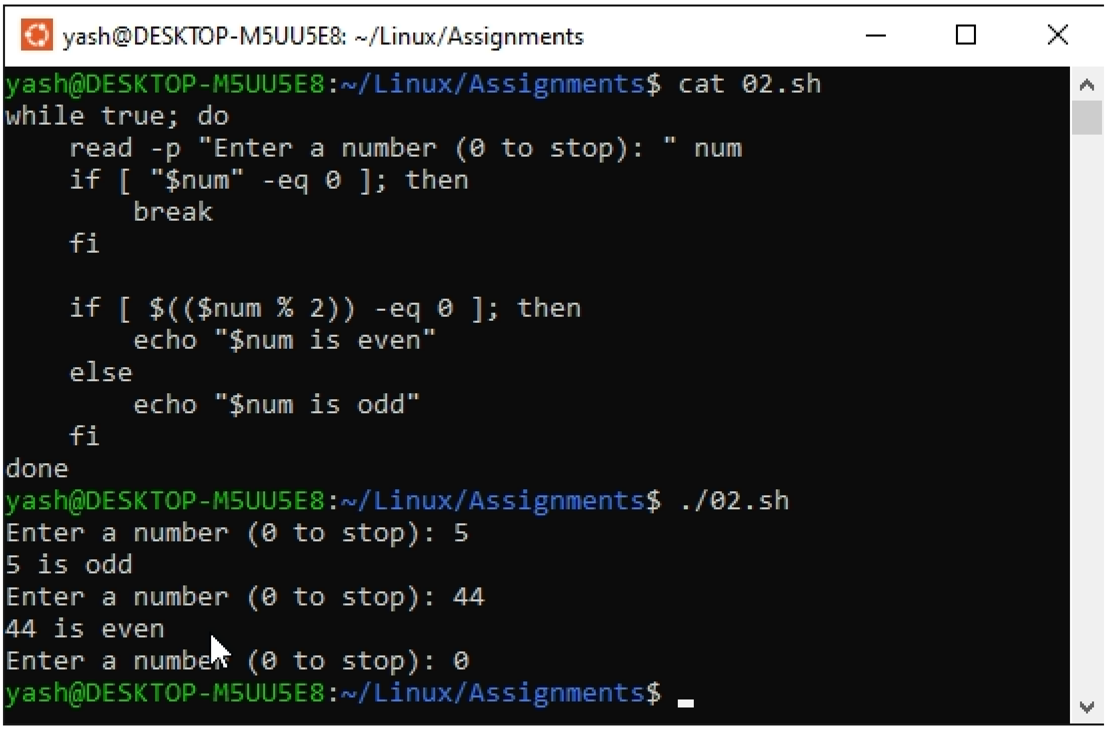
**Assignment 1:** Ensure the script checks if a specific file exists in the current directory. If it exists, print "File exists", otherwise print "File not found".



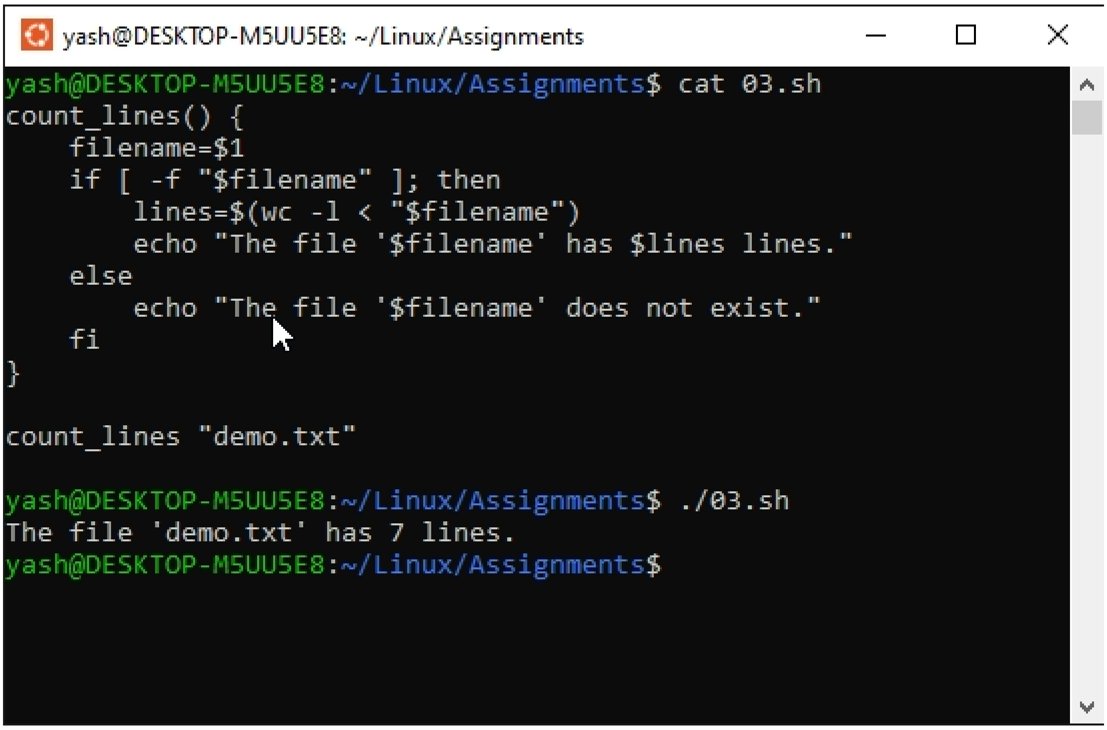
```
yash@DESKTOP-M5UU5E8:~/Linux/Assignments$ cat 01.sh
filename="demo.txt"

if [ -f "$filename" ]; then
    echo "File exists"
else
    echo "File not found"
fi

yash@DESKTOP-M5UU5E8:~/Linux/Assignments$ ./01.sh
File exists
yash@DESKTOP-M5UU5E8:~/Linux/Assignments$
```

**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.



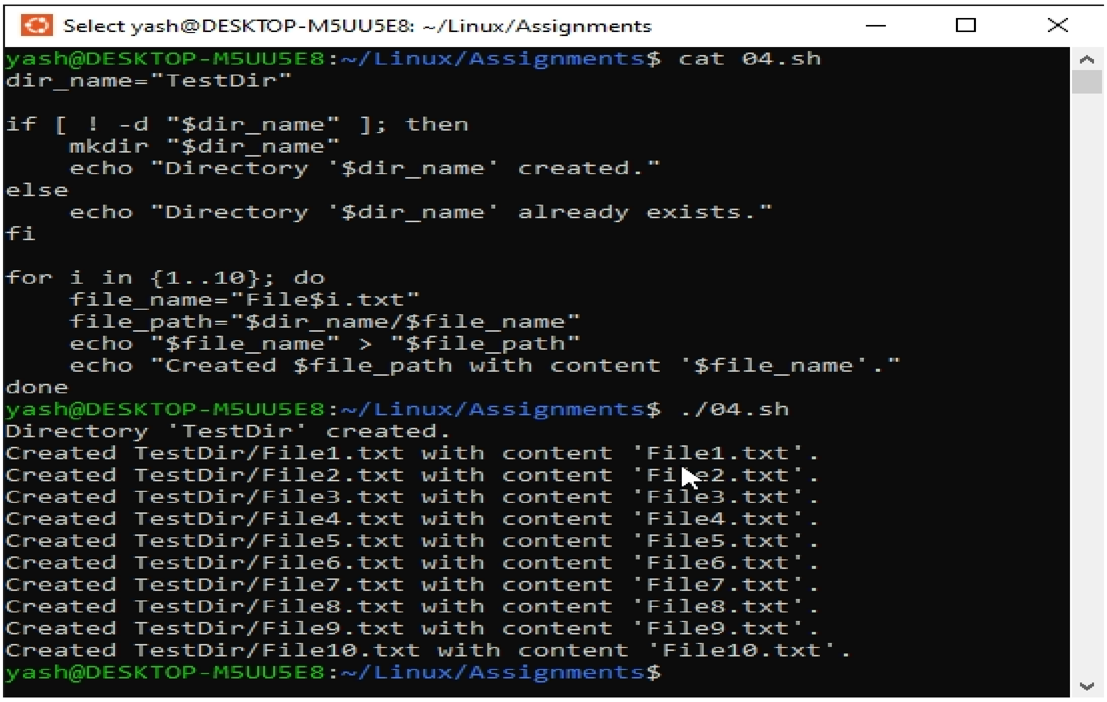
```
yash@DESKTOP-M5UU5E8:~/Linux/Assignments$ cat 02.sh
while true; do
    read -p "Enter a number (0 to stop): " num
    if [ "$num" -eq 0 ]; then
        break
    fi

    if [ $(($num % 2)) -eq 0 ]; then
        echo "$num is even"
    else
        echo "$num is odd"
    fi
done
yash@DESKTOP-M5UU5E8:~/Linux/Assignments$ ./02.sh
Enter a number (0 to stop): 5
5 is odd
Enter a number (0 to stop): 44
44 is even
Enter a number (0 to stop): 0
yash@DESKTOP-M5UU5E8:~/Linux/Assignments$
```

**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.



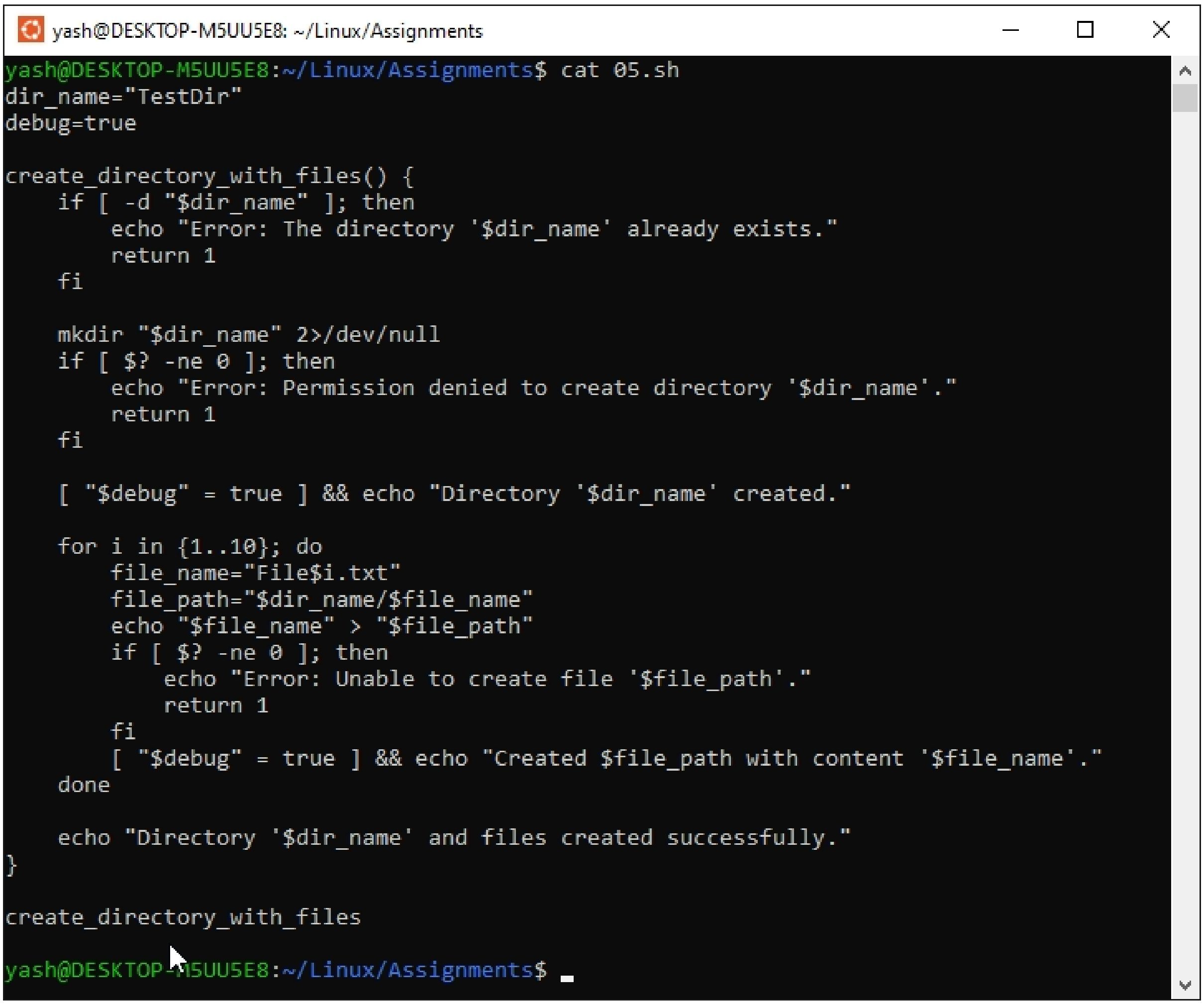
```
yash@DESKTOP-M5UU5E8:~/Linux/Assignments$ cat 03.sh
count_lines() {
    filename=$1
    if [ -f "$filename" ]; then
        lines=$(wc -l < "$filename")
        echo "The file '$filename' has $lines lines."
    else
        echo "The file '$filename' does not exist."
    fi
}

count_lines "demo.txt"

yash@DESKTOP-M5UU5E8:~/Linux/Assignments$ ./03.sh
The file 'demo.txt' has 7 lines.
yash@DESKTOP-M5UU5E8:~/Linux/Assignments$
```

**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").



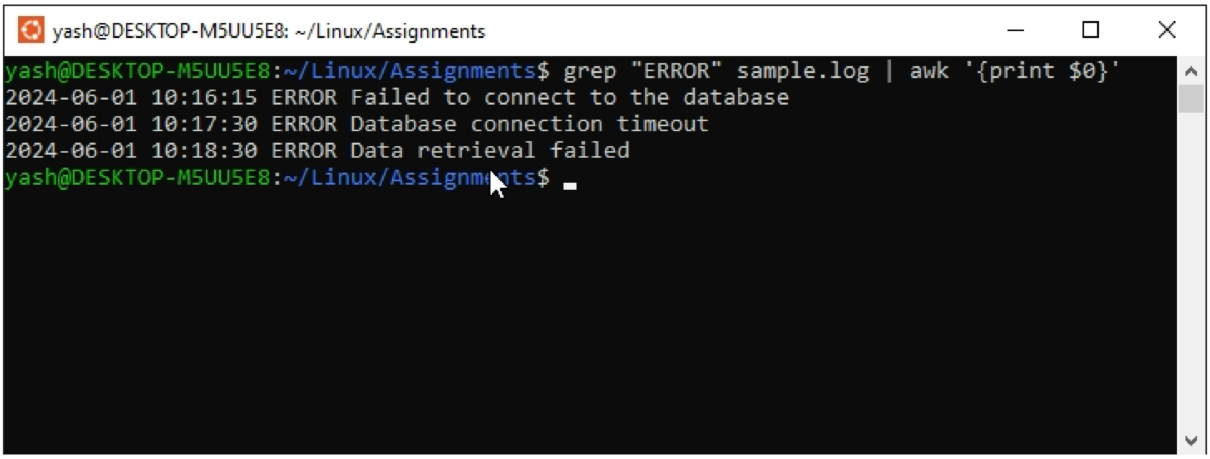
```
yash@DESKTOP-M5UU5E8:~/Linux/Assignments$ cat 04.sh
dir_name="TestDir"

if [ ! -d "$dir_name" ]; then
    mkdir "$dir_name"
    echo "Directory '$dir_name' created."
else
    echo "Directory '$dir_name' already exists."
fi

for i in {1..10}; do
    file_name="File$i.txt"
    file_path="$dir_name/$file_name"
    echo "$file_name" > "$file_path"
    echo "Created $file_path with content '$file_name'."
done
yash@DESKTOP-M5UU5E8:~/Linux/Assignments$ ./04.sh
Directory 'TestDir' created.
Created TestDir/File1.txt with content 'File1.txt'.
Created TestDir/File2.txt with content 'File2.txt'.
Created TestDir/File3.txt with content 'File3.txt'.
Created TestDir/File4.txt with content 'File4.txt'.
Created TestDir/File5.txt with content 'File5.txt'.
Created TestDir/File6.txt with content 'File6.txt'.
Created TestDir/File7.txt with content 'File7.txt'.
Created TestDir/File8.txt with content 'File8.txt'.
Created TestDir/File9.txt with content 'File9.txt'.
Created TestDir/File10.txt with content 'File10.txt'.
yash@DESKTOP-M5UU5E8:~/Linux/Assignments$
```

**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.



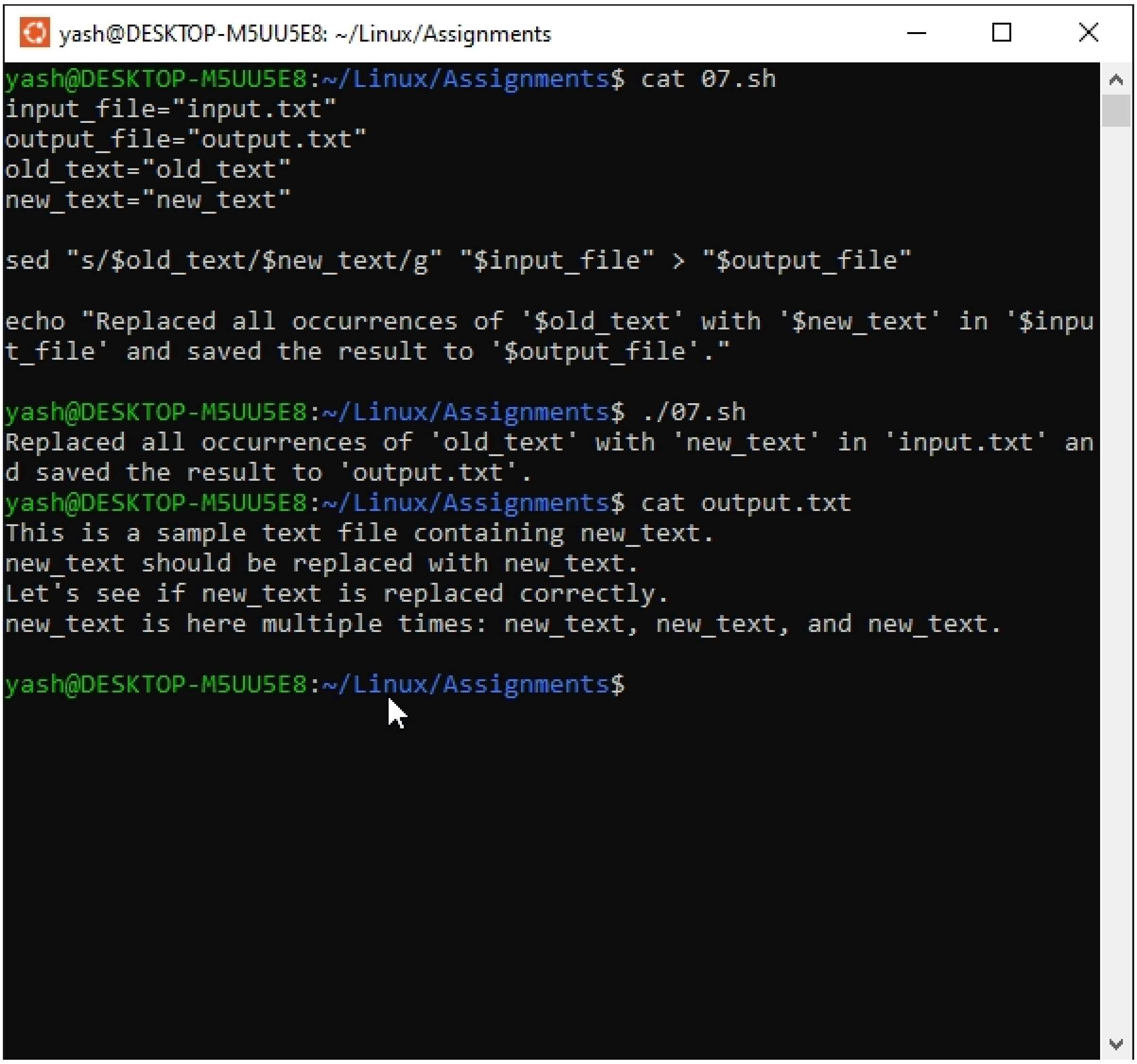
```
yash@DESKTOP-M5UU5E8:~/Linux/Assignments$ cat 05.sh
dir_name="TestDir"
debug=true

create_directory_with_files() {
    if [ -d "$dir_name" ]; then
        echo "Error: The directory '$dir_name' already exists."
        return 1
    fi

    mkdir "$dir_name" 2>/dev/null
    if [ $? -ne 0 ]; then
        echo "Error: Permission denied to create directory '$dir_name'."
        return 1
    fi

    [ "$debug" = true ] && echo "Directory '$dir_name' created."

    for i in {1..10}; do
        file_name="File$i.txt"
        file_path="$dir_name/$file_name"
        echo "$file_name" > "$file_path"
        if [ $? -ne 0 ]; then
            echo "Error: Unable to create file '$file_path'."
            return 1
        fi
        [ "$debug" = true ] && echo "Created $file_path with content '$file_name'."
    done

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

create_directory_with_files

yash@DESKTOP-M5UU5E8:~/Linux/Assignments$ _
```

**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



```
yash@DESKTOP-M5UU5E8:~/Linux/Assignments$ grep "ERROR" sample.log | awk '{print $0}'
2024-06-01 10:16:15 ERROR Failed to connect to the database
2024-06-01 10:17:30 ERROR Database connection timeout
2024-06-01 10:18:30 ERROR Data retrieval failed
yash@DESKTOP-M5UU5E8:~/Linux/Assignments$ _
```

**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.



```
yash@DESKTOP-M5UU5E8:~/Linux/Assignments$ cat 07.sh
input_file="input.txt"
output_file="output.txt"
old_text="old_text"
new_text="new_text"

sed "s/$old_text/$new_text/g" "$input_file" > "$output_file"

echo "Replaced all occurrences of '$old_text' with '$new_text' in '$inpu
t_file' and saved the result to '$output_file'."

yash@DESKTOP-M5UU5E8:~/Linux/Assignments$ ./07.sh
Replaced all occurrences of 'old_text' with 'new_text' in 'input.txt' an
d saved the result to 'output.txt'.
yash@DESKTOP-M5UU5E8:~/Linux/Assignments$ cat output.txt
This is a sample text file containing new_text.
new_text should be replaced with new_text.
Let's see if new_text is replaced correctly.
new_text is here multiple times: new_text, new_text, and new_text.

yash@DESKTOP-M5UU5E8:~/Linux/Assignments$
```