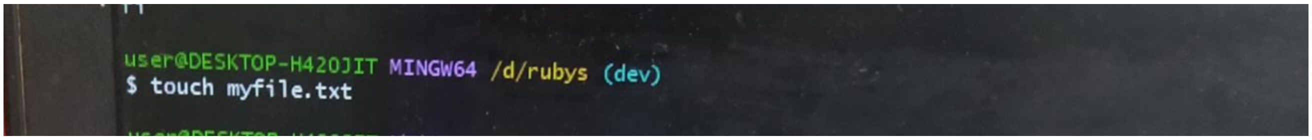
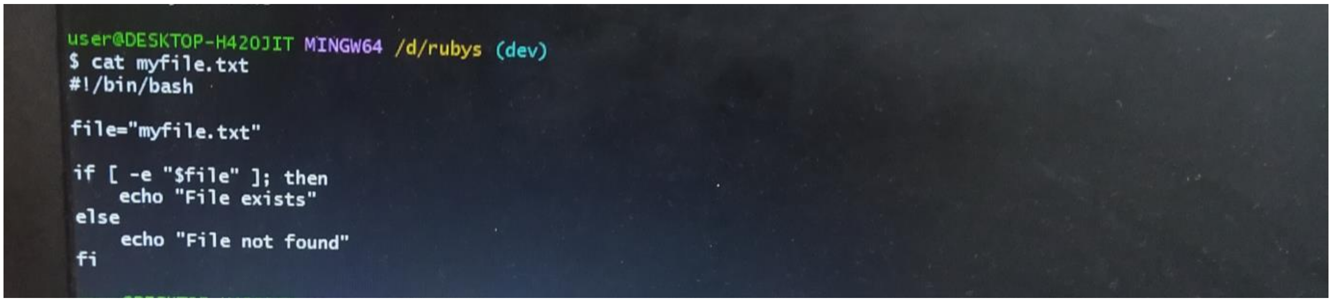
ASSIGNMNET_06: SHELL SCRIPTING WITH BASH

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".

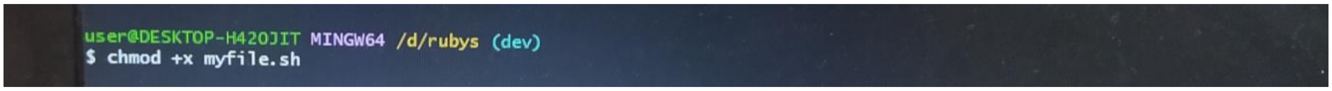
ANS:



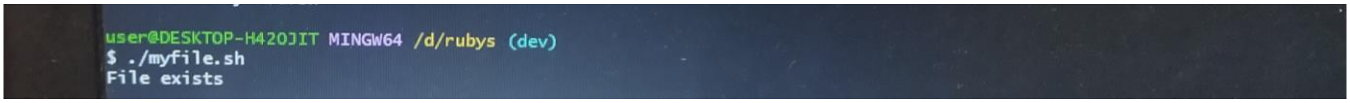
```
user@DESKTOP-H420JIT MINGW64 /d/rubys (dev)
$ touch myfile.txt
```



```
user@DESKTOP-H420JIT MINGW64 /d/rubys (dev)
$ cat myfile.txt
#!/bin/bash

file="myfile.txt"

if [ -e "$file" ]; then
    echo "File exists"
else
    echo "File not found"
fi
```



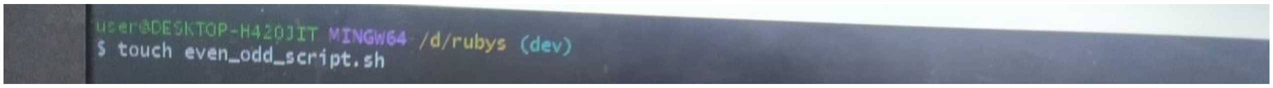
```
user@DESKTOP-H420JIT MINGW64 /d/rubys (dev)
$ chmod +x myfile.sh
```



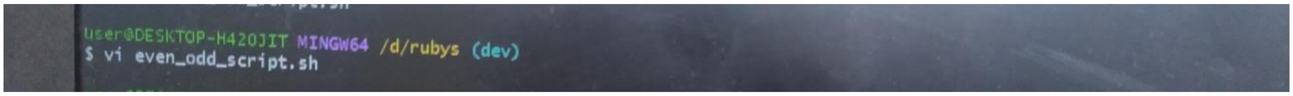
```
user@DESKTOP-H420JIT MINGW64 /d/rubys (dev)
$ ./myfile.sh
File exists
```

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.

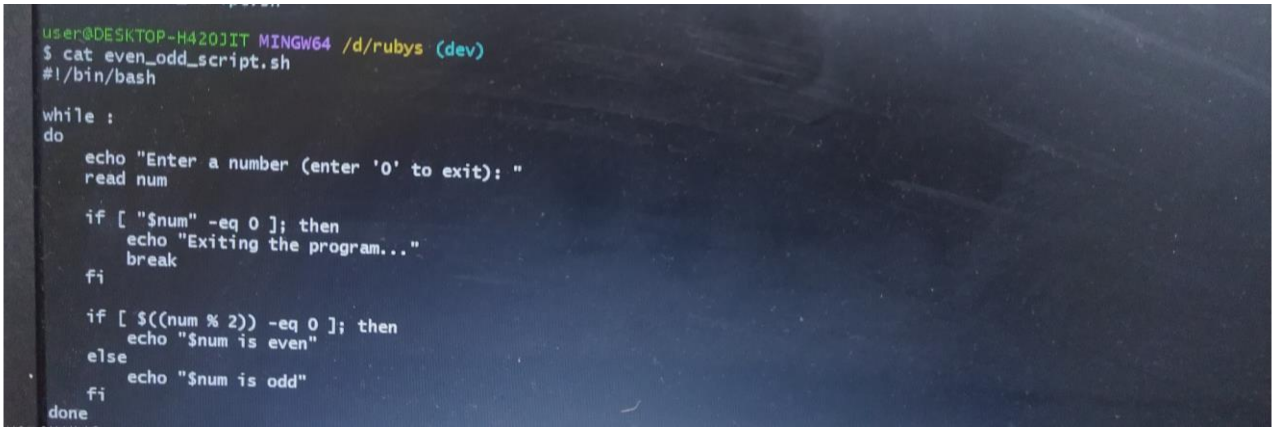
ANS:



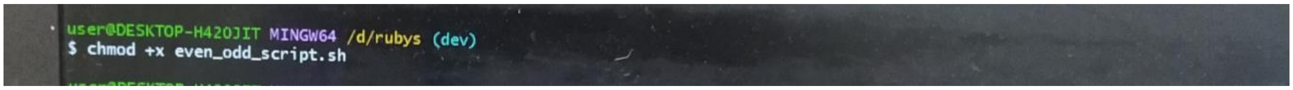
```
user@DESKTOP-H420JIT MINGW64 /d/rubys (dev)
$ touch even_odd_script.sh
```



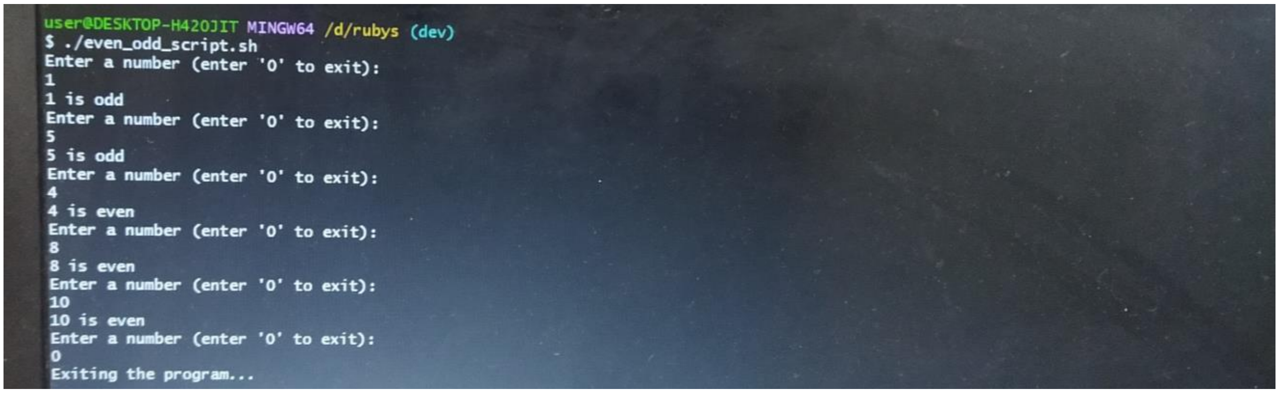
```
user@DESKTOP-H420JIT MINGW64 /d/rubys (dev)
$ vi even_odd_script.sh
```



```
user@DESKTOP-H420JIT MINGW64 /d/rubys (dev)
$ cat even_odd_script.sh
#!/bin/bash

while :
do
    echo "Enter a number (enter '0' to exit): "
    read num

    if [ "$num" -eq 0 ]; then
        echo "Exiting the program..."
        break
    fi

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



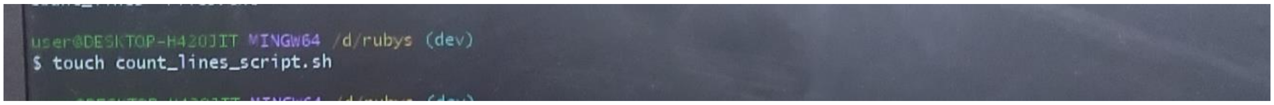
```
user@DESKTOP-H420JIT MINGW64 /d/rubys (dev)
$ chmod +x even_odd_script.sh
```



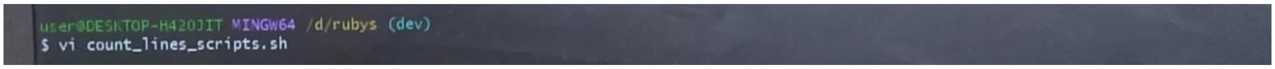
```
user@DESKTOP-H420JIT MINGW64 /d/rubys (dev)
$ ./even_odd_script.sh
Enter a number (enter '0' to exit):
1
1 is odd
Enter a number (enter '0' to exit):
5
5 is odd
Enter a number (enter '0' to exit):
4
4 is even
Enter a number (enter '0' to exit):
8
8 is even
Enter a number (enter '0' to exit):
10
10 is even
Enter a number (enter '0' to exit):
0
Exiting the program...
```

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.

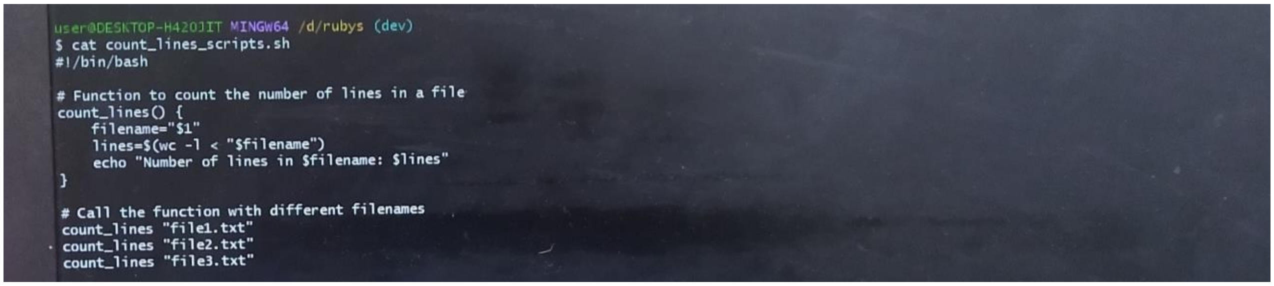
ANS:



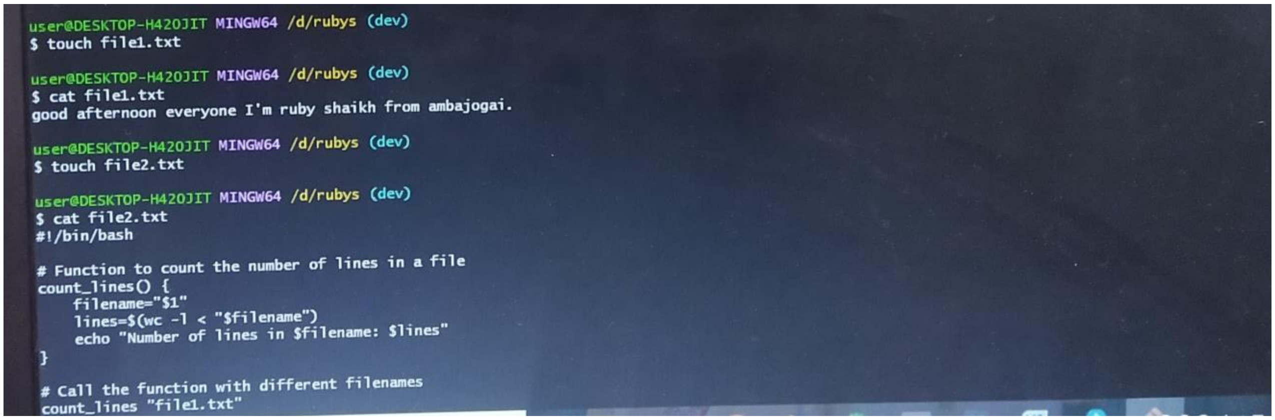
```
user@DESKTOP-H420JIT MINGW64 /d/rubys (dev)
$ touch count_lines_script.sh
```



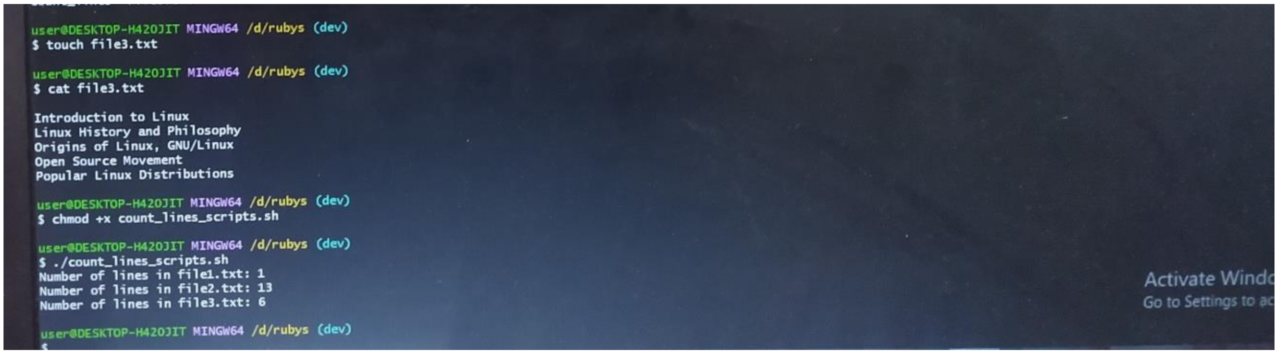
```
user@DESKTOP-H420JIT MINGW64 /d/rubys (dev)
$ vi count_lines_scripts.sh
```



```
user@DESKTOP-H420JIT MINGW64 /d/rubys (dev)
$ cat count_lines_scripts.sh
#!/bin/bash

# Function to count the number of lines in a file
count_lines() {
    filename="$1"
    lines=$(wc -l < "$filename")
    echo "Number of lines in $filename: $lines"
}

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



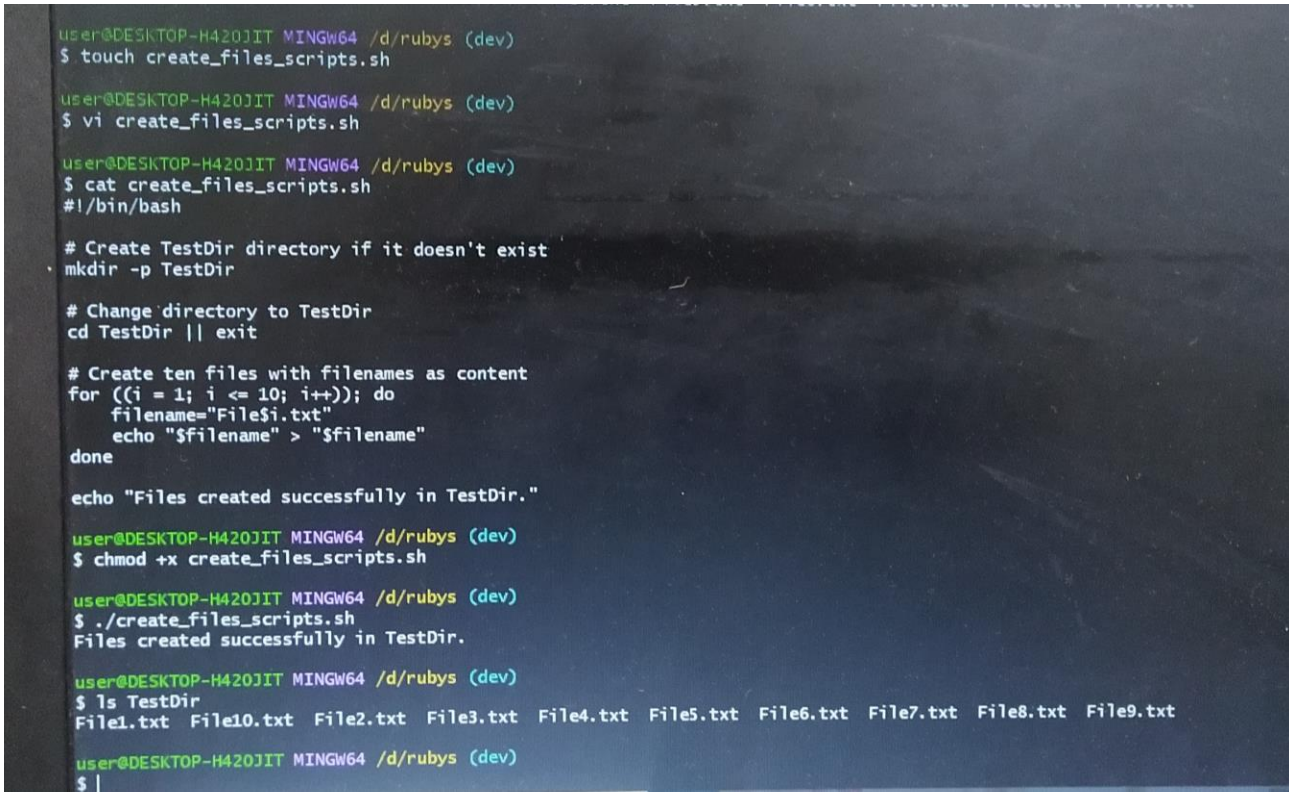
```
user@DESKTOP-H420JIT MINGW64 /d/rubys (dev)
$ touch file1.txt

user@DESKTOP-H420JIT MINGW64 /d/rubys (dev)
$ cat file1.txt
good afternoon everyone I'm ruby shaikh from ambajogai.

user@DESKTOP-H420JIT MINGW64 /d/rubys (dev)
$ touch file2.txt

user@DESKTOP-H420JIT MINGW64 /d/rubys (dev)
$ cat file2.txt
#!/bin/bash

# Function to count the number of lines in a file
count_lines() {
    filename="$1"
    lines=$(wc -l < "$filename")
    echo "Number of lines in $filename: $lines"
}

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



```
user@DESKTOP-H420JIT MINGW64 /d/rubys (dev)
$ touch file3.txt

user@DESKTOP-H420JIT MINGW64 /d/rubys (dev)
$ cat file3.txt

Introduction to Linux
Linux History and Philosophy
Origins of Linux, GNU/Linux
Open Source Movement
Popular Linux Distributions

user@DESKTOP-H420JIT MINGW64 /d/rubys (dev)
$ chmod +x count_lines_scripts.sh

user@DESKTOP-H420JIT MINGW64 /d/rubys (dev)
$ ./count_lines_scripts.sh
Number of lines in file1.txt: 1
Number of lines in file2.txt: 13
Number of lines in file3.txt: 6

user@DESKTOP-H420JIT MINGW64 /d/rubys (dev)
$
```

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").

ANS:

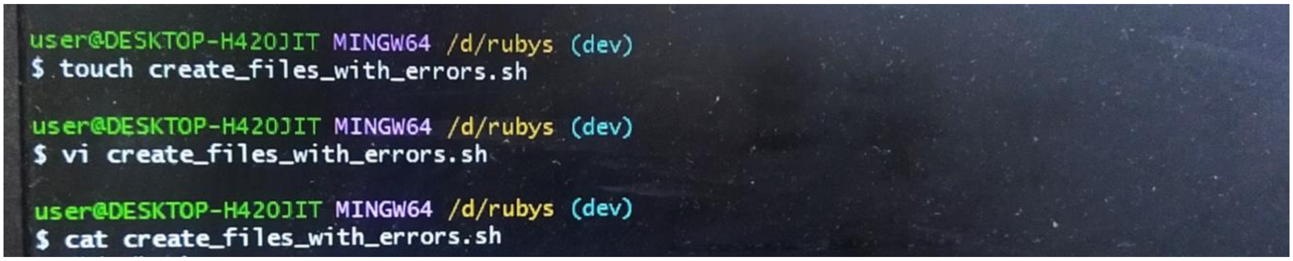


```
user@DESKTOP-H420JIT MINGW64 /d/rubys (dev)
$ touch create_files_scripts.sh

user@DESKTOP-H420JIT MINGW64 /d/rubys (dev)
$ vi create_files_scripts.sh

user@DESKTOP-H420JIT MINGW64 /d/rubys (dev)
$ cat create_files_scripts.sh
#!/bin/bash

# Create TestDir directory if it doesn't exist
mkdir -p TestDir

# Change directory to TestDir
cd TestDir || exit

# Create ten files with filenames as content
for ((i = 1; i <= 10; i++)); do
    filename="File$i.txt"
    echo "$filename" > "$filename"
done

echo "Files created successfully in TestDir."

user@DESKTOP-H420JIT MINGW64 /d/rubys (dev)
$ chmod +x create_files_scripts.sh

user@DESKTOP-H420JIT MINGW64 /d/rubys (dev)
$ ./create_files_scripts.sh
Files created successfully in TestDir.

user@DESKTOP-H420JIT MINGW64 /d/rubys (dev)
$ ls TestDir
File1.txt  File10.txt  File2.txt  File3.txt  File4.txt  File5.txt  File6.txt  File7.txt  File8.txt  File9.txt

user@DESKTOP-H420JIT MINGW64 /d/rubys (dev)
$ |
```

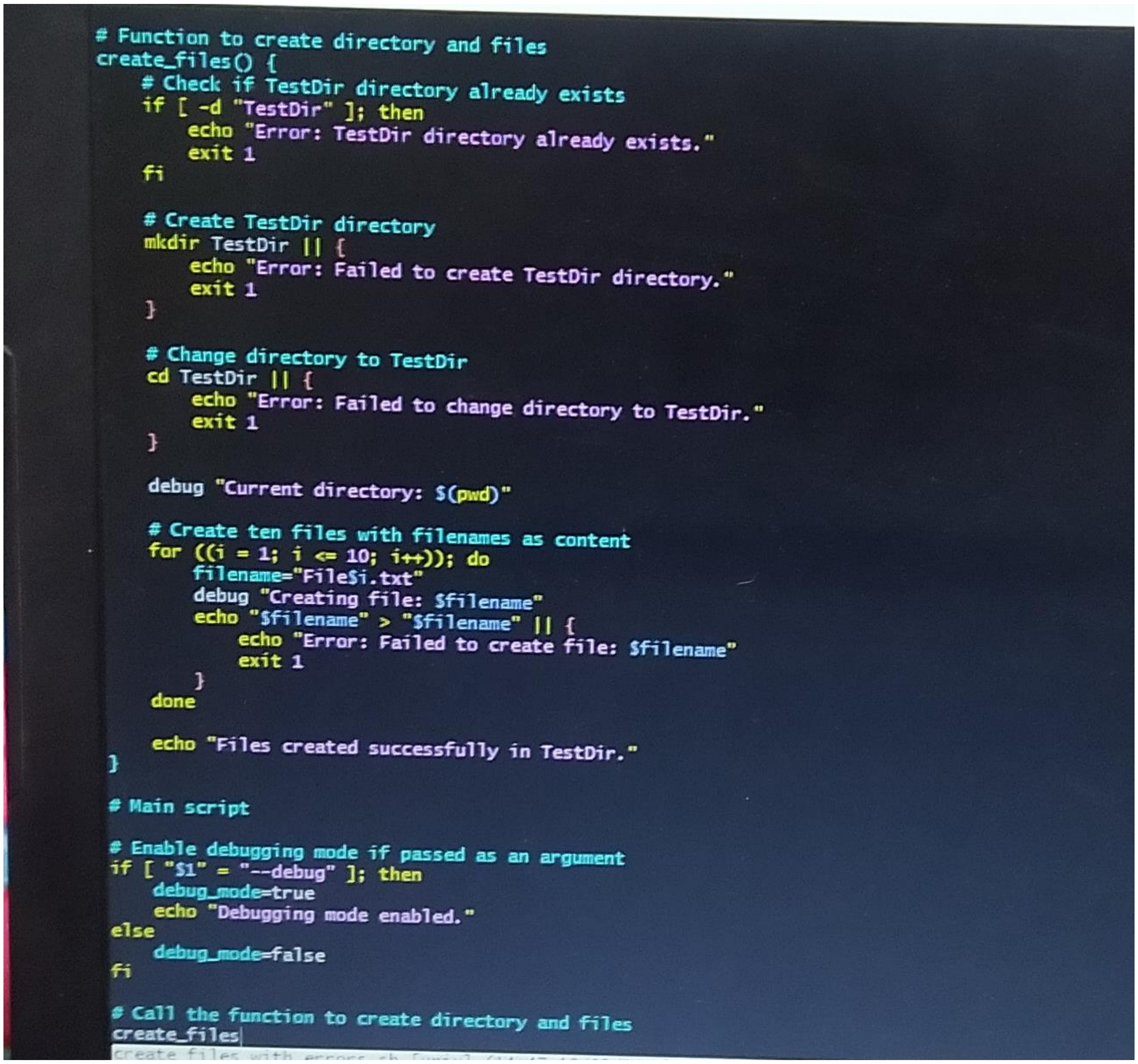
Assignment 5: Modify the script to handle errors, such as the directory already existing or lacking permissions to create files.

Add a debugging mode that prints additional information when enabled.

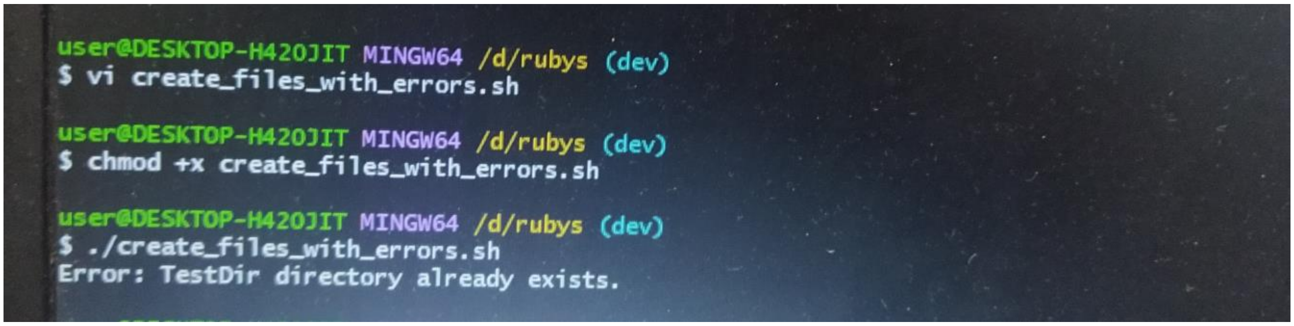
ANS:



```
user@DESKTOP-H420JIT MINGW64 /d/rubys (dev)
$ touch create_files_with_errors.sh

user@DESKTOP-H420JIT MINGW64 /d/rubys (dev)
$ vi create_files_with_errors.sh

user@DESKTOP-H420JIT MINGW64 /d/rubys (dev)
$ cat create_files_with_errors.sh
```



```
# Function to create directory and files
create_files() {
    # Check if TestDir directory already exists
    if [ -d "TestDir" ]; then
        echo "Error: TestDir directory already exists."
        exit 1
    fi

    # Create TestDir directory
    mkdir TestDir || {
        echo "Error: Failed to create TestDir directory."
        exit 1
    }

    # Change directory to TestDir
    cd TestDir || {
        echo "Error: Failed to change directory to TestDir."
        exit 1
    }

    debug "Current directory: $(pwd)"

    # Create ten files with filenames as content
    for ((i = 1; i <= 10; i++)); do
        filename="File$i.txt"
        debug "Creating file: $filename"
        echo "$filename" > "$filename" || {
            echo "Error: Failed to create file: $filename"
            exit 1
        }
    done

    echo "Files created successfully in TestDir."
}

# Main script

# Enable debugging mode if passed as an argument
if [ "$1" = "--debug" ]; then
    debug_mode=true
    echo "Debugging mode enabled."
else
    debug_mode=false
fi

# Call the function to create directory and files
create_files
```

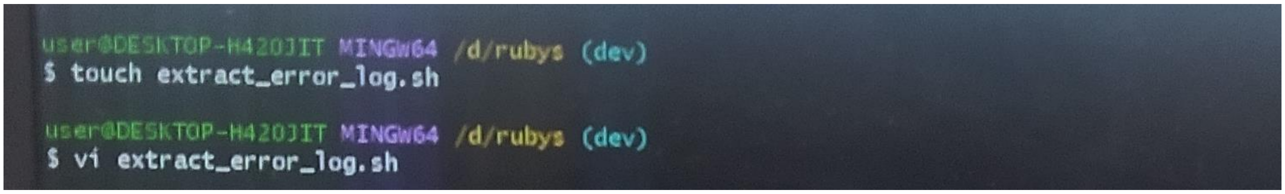


```
user@DESKTOP-H420JIT MINGW64 /d/rubys (dev)
$ vi create_files_with_errors.sh

user@DESKTOP-H420JIT MINGW64 /d/rubys (dev)
$ chmod +x create_files_with_errors.sh

user@DESKTOP-H420JIT MINGW64 /d/rubys (dev)
$ ./create_files_with_errors.sh
Error: TestDir directory already exists.
```

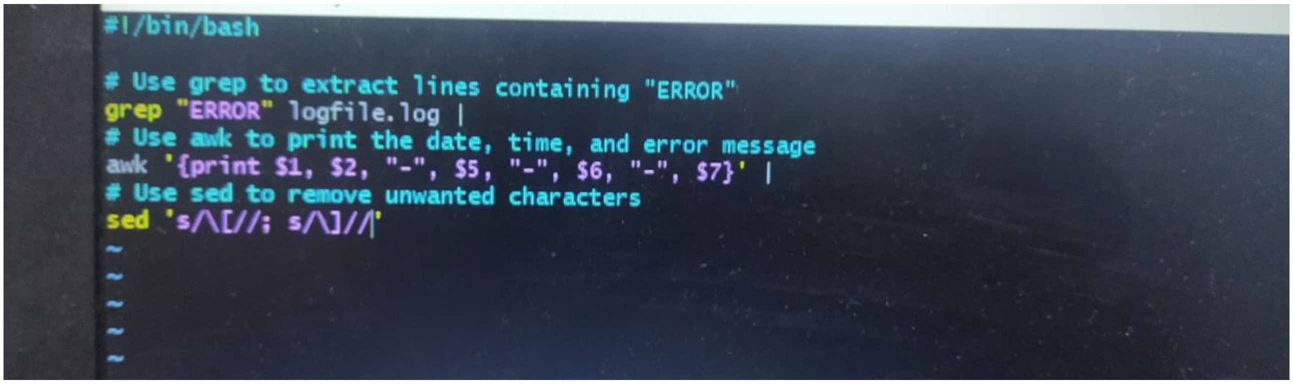
Assignment 6: Given a sample log file, write a script using grep to extract all lines containing "ERROR". Use awk to print the date, time, and error message of each extracted line.

Data Processing with sed

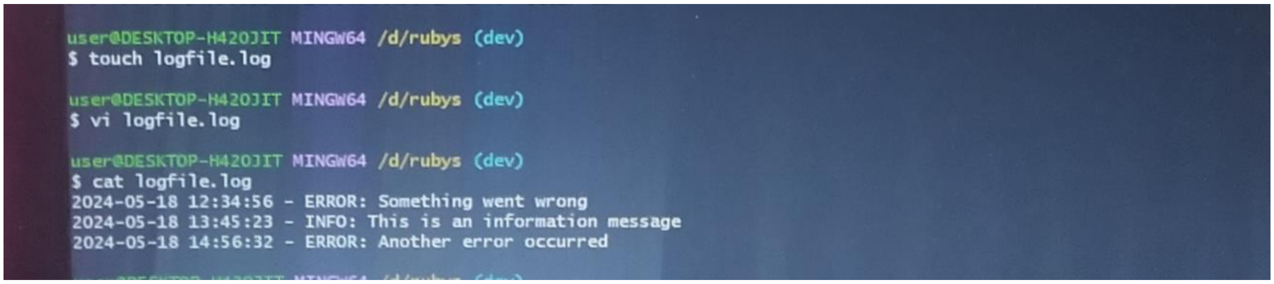
ANS:



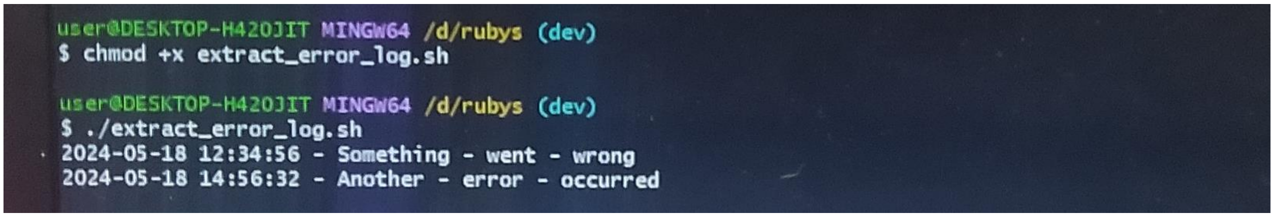
```
user@DESKTOP-H420JIT MINGW64 /d/rubys (dev)
$ touch extract_error_log.sh

user@DESKTOP-H420JIT MINGW64 /d/rubys (dev)
$ vi extract_error_log.sh
```



```
#!/bin/bash

# Use grep to extract lines containing "ERROR"
grep "ERROR" logfile.log |
# Use awk to print the date, time, and error message
awk '{print $1, $2, "-", $5, "-", $6, "-", $7}' |
# Use sed to remove unwanted characters
sed 's/\[//; s/\]//'
~
~
~
~
~
```



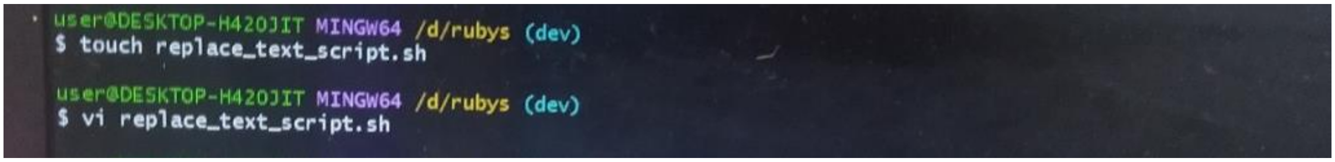
```
user@DESKTOP-H420JIT MINGW64 /d/rubys (dev)
$ touch logfile.log

user@DESKTOP-H420JIT MINGW64 /d/rubys (dev)
$ vi logfile.log

user@DESKTOP-H420JIT MINGW64 /d/rubys (dev)
$ cat logfile.log
2024-05-18 12:34:56 - ERROR: Something went wrong
2024-05-18 13:45:23 - INFO: This is an information message
2024-05-18 14:56:32 - ERROR: Another error occurred

user@DESKTOP-H420JIT MINGW64 /d/rubys (dev)
```



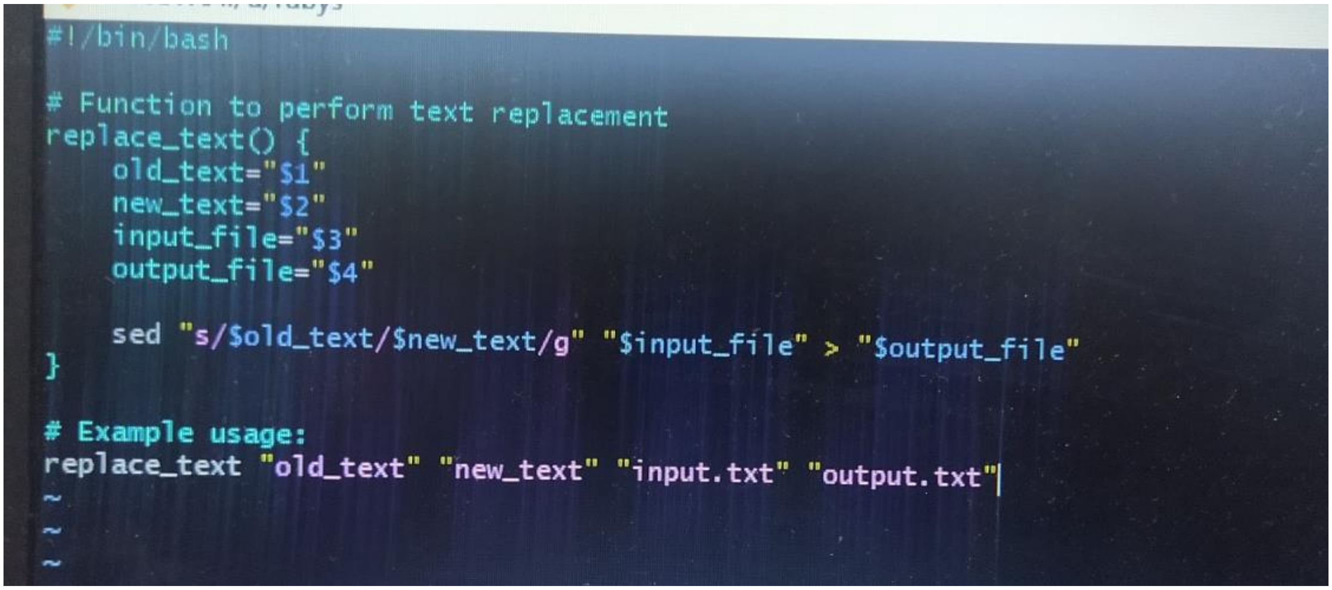
```
user@DESKTOP-H420JIT MINGW64 /d/rubys (dev)
$ chmod +x extract_error_log.sh

user@DESKTOP-H420JIT MINGW64 /d/rubys (dev)
$ ./extract_error_log.sh
2024-05-18 12:34:56 - Something - went - wrong
2024-05-18 14:56:32 - Another - error - occurred
```

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.

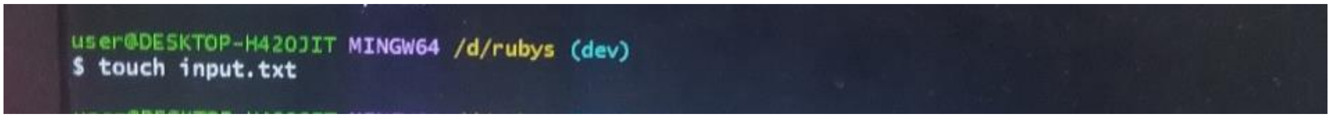
ANS:



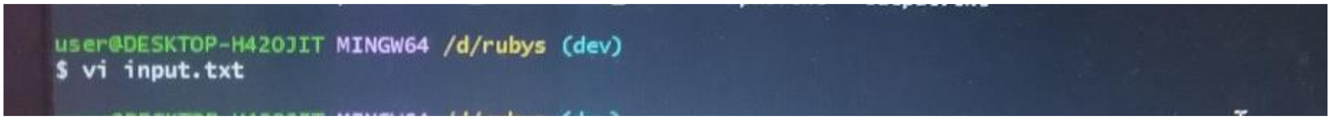
```
user@DESKTOP-H420JIT MINGW64 /d/rubys (dev)
$ touch replace_text_script.sh

user@DESKTOP-H420JIT MINGW64 /d/rubys (dev)
$ vi replace_text_script.sh
```



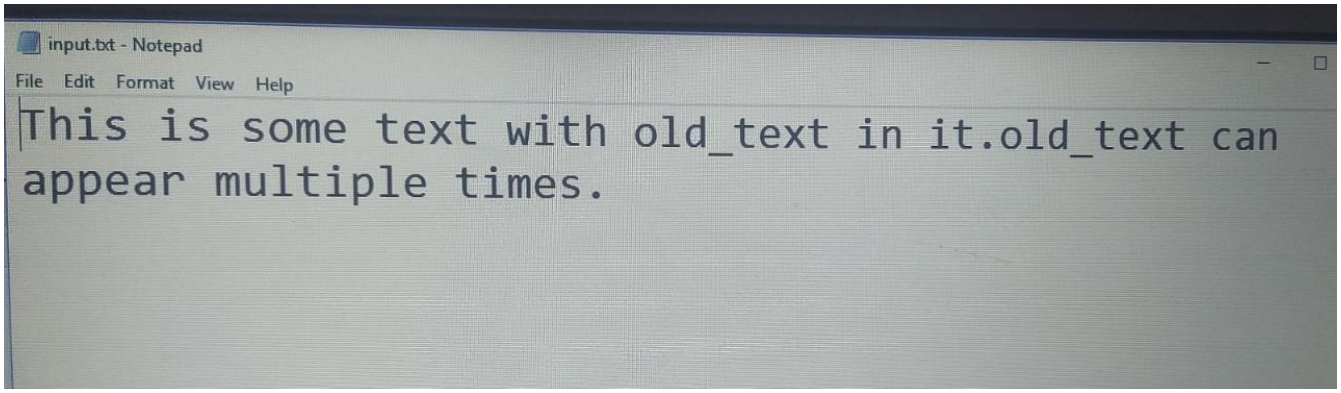
```
#!/bin/bash

# Function to perform text replacement
replace_text() {
    old_text="$1"
    new_text="$2"
    input_file="$3"
    output_file="$4"

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

# Example usage:
replace_text "old_text" "new_text" "input.txt" "output.txt"
~
~
~
```



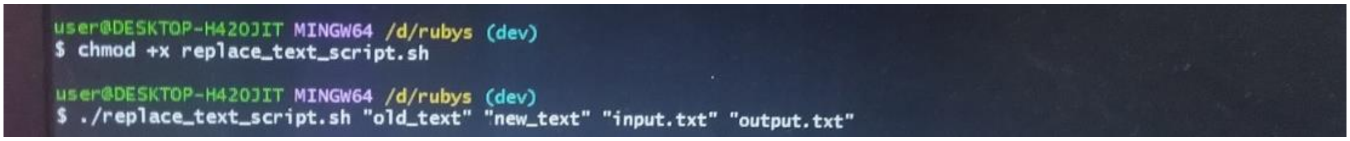
```
user@DESKTOP-H420JIT MINGW64 /d/rubys (dev)
$ touch input.txt
```



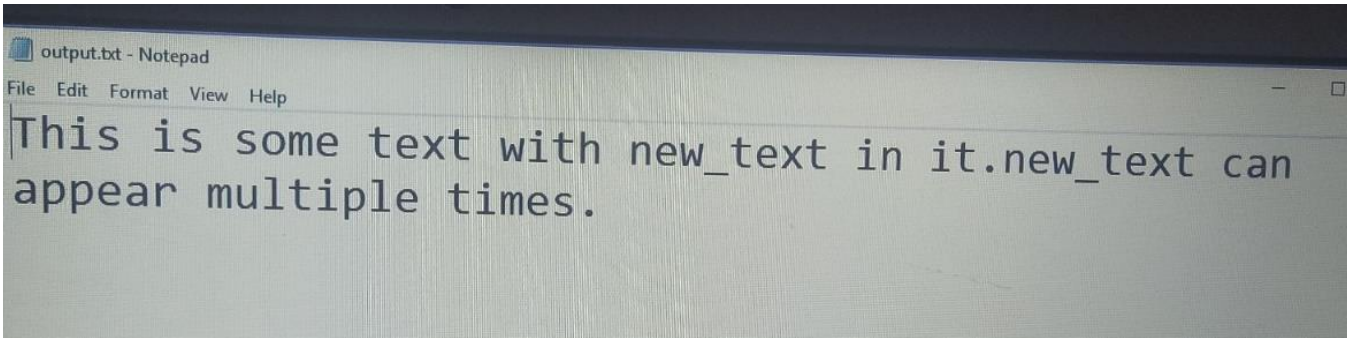
```
user@DESKTOP-H420JIT MINGW64 /d/rubys (dev)
$ vi input.txt
```



```
This is some text with old_text in it.old_text can
appear multiple times.
```



```
user@DESKTOP-H42OJIT MINGW64 /d/rubys (dev)
$ chmod +x replace_text_script.sh

user@DESKTOP-H42OJIT MINGW64 /d/rubys (dev)
$ ./replace_text_script.sh "old_text" "new_text" "input.txt" "output.txt"
```



```
This is some text with new_text in it.new_text can
appear multiple times.
```