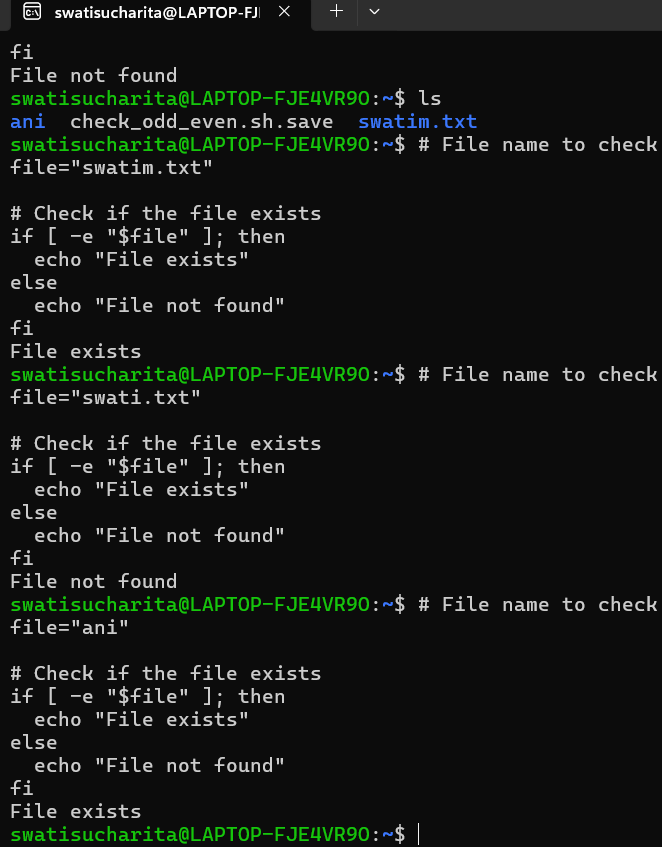
**Assignment 1: Ensure the script checks if a specific file (e.g., myfile.txt) exists in the current directory. If it exists, print "File exists", otherwise print "File not found".**

**Answer:**

# File name to check  
file="swatim.txt"

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

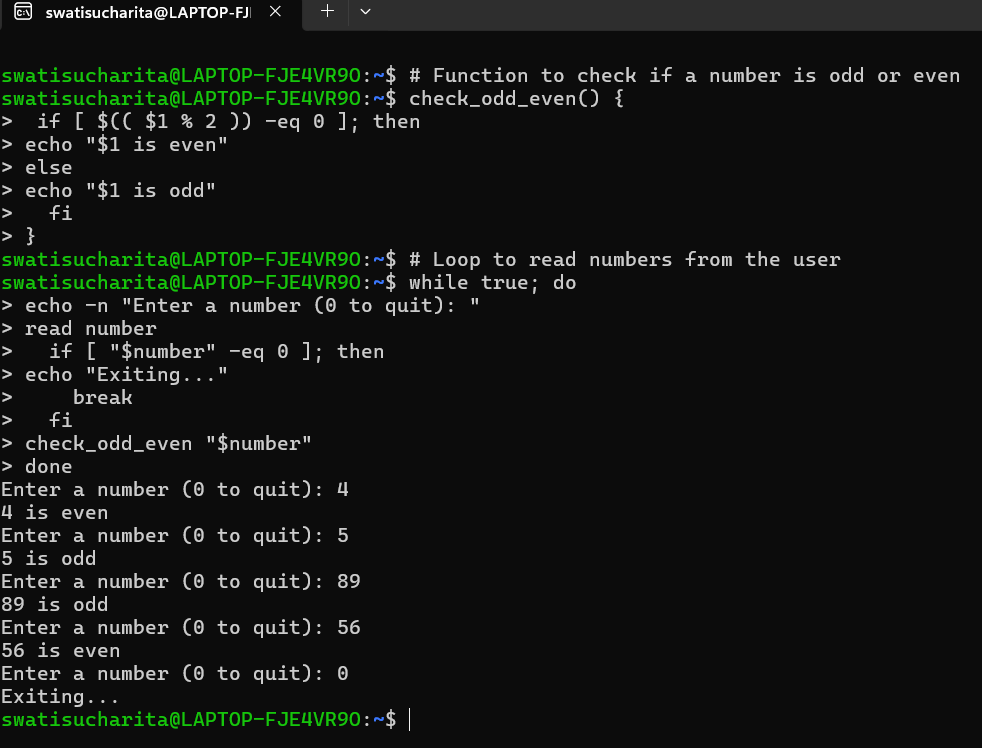


**Assignment 2: Write a script that reads numbers from the user until they enter '0'. The script should also print whether each number is odd or even.**

**Answer:**

# Function to check if a number is odd or even  
check\_odd\_even() {  
  if [ $(( $1 % 2 )) -eq 0 ]; then  
    echo "$1 is even"  
  else  
    echo "$1 is odd"  
  fi  
}

# Loop to read numbers from the user  
while true; do  
  echo -n "Enter a number (0 to quit): "  
  read number  
  
  if [ "$number" -eq 0 ]; then  
    echo "Exiting..."  
    break  
  fi  
  
  check\_odd\_even "$number"  
done



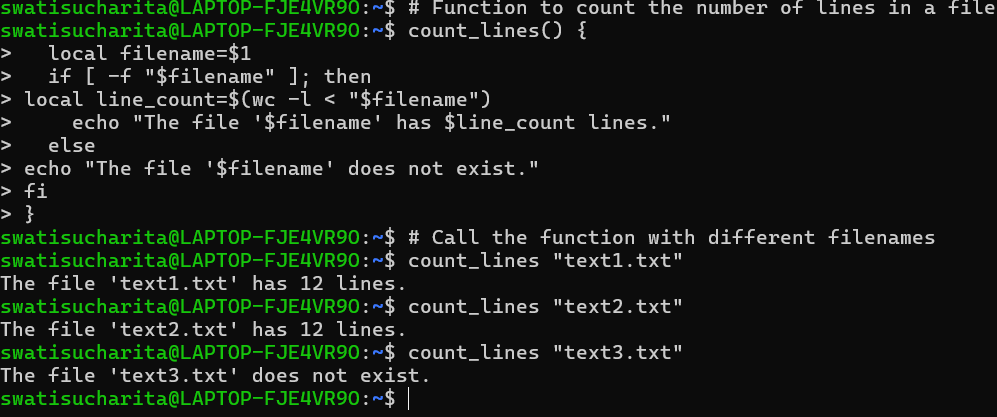
**Assignment 3: Create a function that takes a filename as an argument and prints the number of lines in the file. Call this function from your script with different filenames.**

**Answer:**

# Function to count the number of lines in a file  
count\_lines() {  
  local filename=$1

  if [ -f "$filename" ]; then  
    local line\_count=$(wc -l < "$filename")  
    echo "The file '$filename' has $line\_count lines."  
  else  
    echo "The file '$filename' does not exist."  
  fi  
}

# Call the function with different filenames  
count\_lines "file1.txt"  
count\_lines "file2.txt"  
count\_lines "file3.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").**

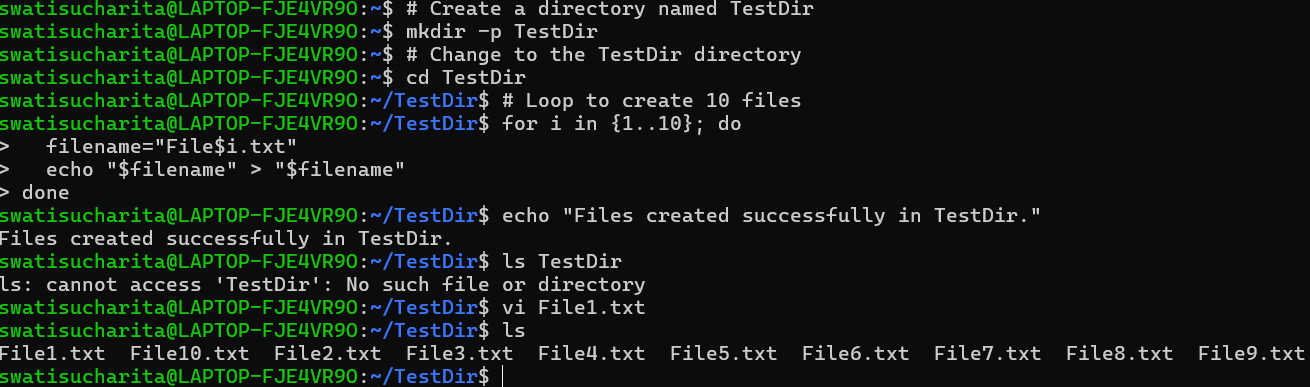
**Answer:**

# Create a directory named TestDir  
mkdir -p TestDir

# Change to the TestDir directory  
cd TestDir

# Loop to create 10 files  
for i in {1..10}; do  
  filename="File$i.txt"  
  echo "$filename" > "$filename"  
done

echo "Files created successfully in 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.**

**Answer:**

# Set debugging mode (0 for off, 1 for on)  
DEBUG=1

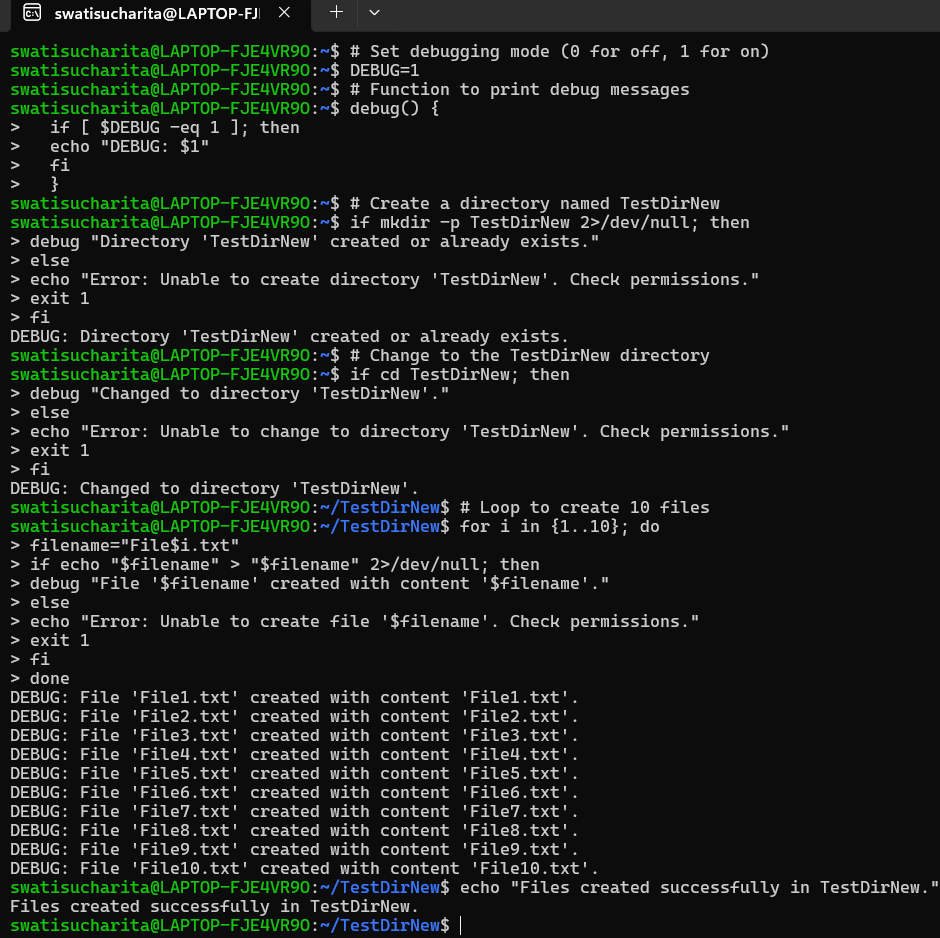
# Function to print debug messages  
debug() {  
  if [ $DEBUG -eq 1 ]; then  
    echo "DEBUG: $1"  
  fi  
}

# Create a directory named TestDirNew  
if mkdir -p TestDirNew 2>/dev/null; then  
  debug "Directory 'TestDirNew' created or already exists."  
else  
  echo "Error: Unable to create directory 'TestDirNew'. Check permissions."  
  exit 1  
fi

# Change to the TestDirNew directory  
if cd TestDirNew; then  
  debug "Changed to directory 'TestDirNew'."  
else  
  echo "Error: Unable to change to directory 'TestDirNew'. Check permissions."  
  exit 1  
fi

# Loop to create 10 files  
for i in {1..10}; do  
  filename="File$i.txt"  
  if echo "$filename" > "$filename" 2>/dev/null; then  
    debug "File '$filename' created with content '$filename'."  
  else  
    echo "Error: Unable to create file '$filename'. Check permissions."  
    exit 1  
  fi  
done

echo "Files created successfully in TestDirNew."



**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**

**Answer:**

mkdir log\_processing

cd log\_processing

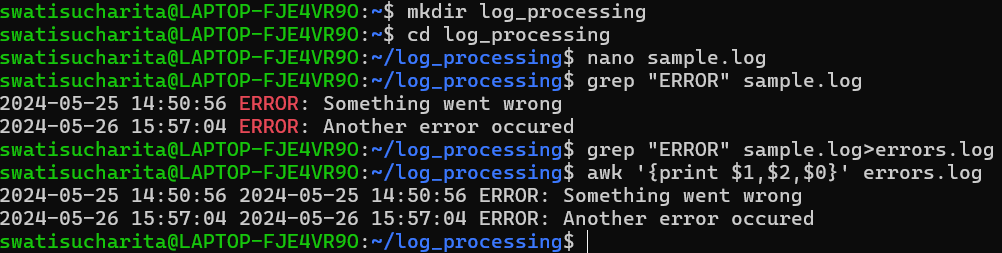
nano sample.log

2024-05-27 12:34:56 ERROR: Something went wrong

2024-05-28 13:45:57 ERROR: Another error occurred

grep "ERROR" sample.log > errors.log

awk '{print $1, $2, $0}' errors.log



**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.**

**Answer:**

# Check if the correct number of arguments is provided  
if [ "$#" -ne 4 ]; then  
    echo "Usage: $0 input\_file old\_text new\_text output\_file"  
   #exit 1  
fi

# Assign input arguments to variables  
INPUT\_FILE=$1  
OLD\_TEXT=$2  
NEW\_TEXT=$3  
OUTPUT\_FILE=$4

# Use sed to replace all occurrences of old\_text with new\_text  
sed "s/$OLD\_TEXT/$NEW\_TEXT/g" "$INPUT\_FILE" > "$OUTPUT\_FILE"

# Print a message indicating the operation is complete  
echo "Replaced all occurrences of '$OLD\_TEXT' with '$NEW\_TEXT' in '$INPUT\_FILE' and saved

