ASSIGNMENT:2

ADITYA MISHRA, BATCH-78 ,SAP.ID.590029219

Q1.Write a script that adds a user-defined prefix or suffix to all files in a directory.

1. Create the Script File

Open a terminal and create a new file: `nano rename_files.sh`

2. Paste the Following Bash Script

```
#!/bin/bash

# Ask for directory
read -p "Enter target directory: " directory

# Validate directory
if [[ ! -d "$directory" ]]; then
    echo "Error: Directory does not exist."
    exit 1
fi

# Ask for prefix and suffix
read -p "Enter prefix (leave blank for none): " prefix
read -p "Enter suffix (leave blank for none): " suffix

# Loop through files
for file in "$directory"/*; do
    # Skip if no files exist
    [[ -e "$file" ]] || continue

    # Skip directories
    if [[ -d "$file" ]]; then
        continue
    fi

    filename=$(basename "$file")
    name="${filename%.*}"     # file name without extension
    ext="${filename##*.}"     # extension

    # If file has no extension
    if [[ "$filename" == "$ext" ]]; then
        newname="${prefix}${name}${suffix}"
    else
        newname="${prefix}${name}${suffix}.${ext}"
    fi

    mv "$file" "$directory/$newname"
    echo "Renamed: $filename → $newname"
done
```

```
echo "Done!"
```

3. Save and Exit

In nano: CTRL + O (write) ENTER (confirm) CTRL + X (exit)

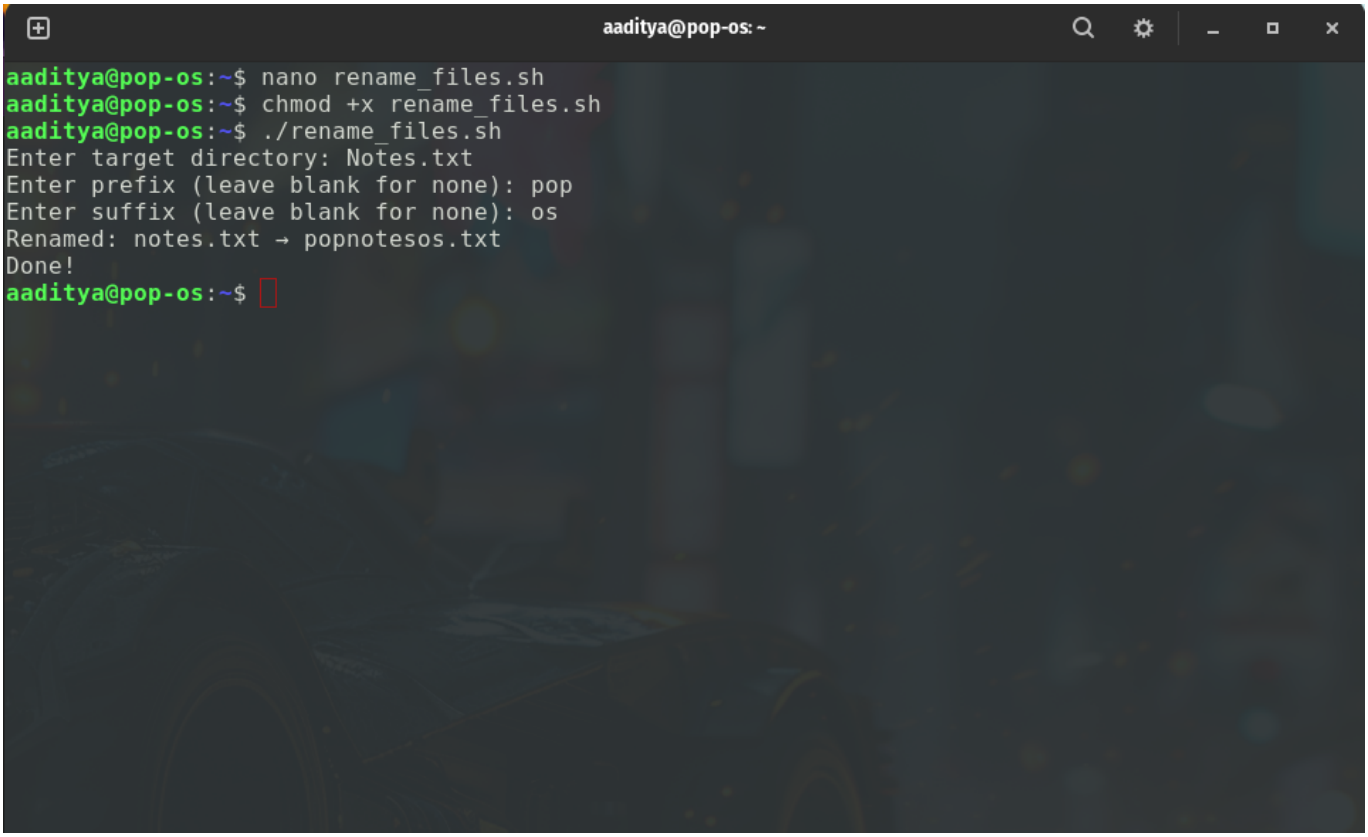
4. Make the Script Executable

```
chmod +x rename_files.sh
```

5. Run the Script

```
./rename_files.sh
```

Output:



Q2.Search recursively for files with a given extension or larger than a specified size.

1. Overview

The find command is one of the most powerful tools in Linux for searching directories recursively.

You can use it to search by:

- File extension
- File size

- File type
- Name patterns
- And much more

2. Search for Files by Extension

Step 1 — Choose your starting directory

Example: search inside the current directory: `cd /path/to/start/directory` Step 2 — Run the find command Syntax `find <directory> -type f -name "*.extension"`

3. Search for Files Larger Than a Certain Size

Step 1 — Pick your size unit

find supports:

k → kilobytes

M → megabytes

G → gigabytes

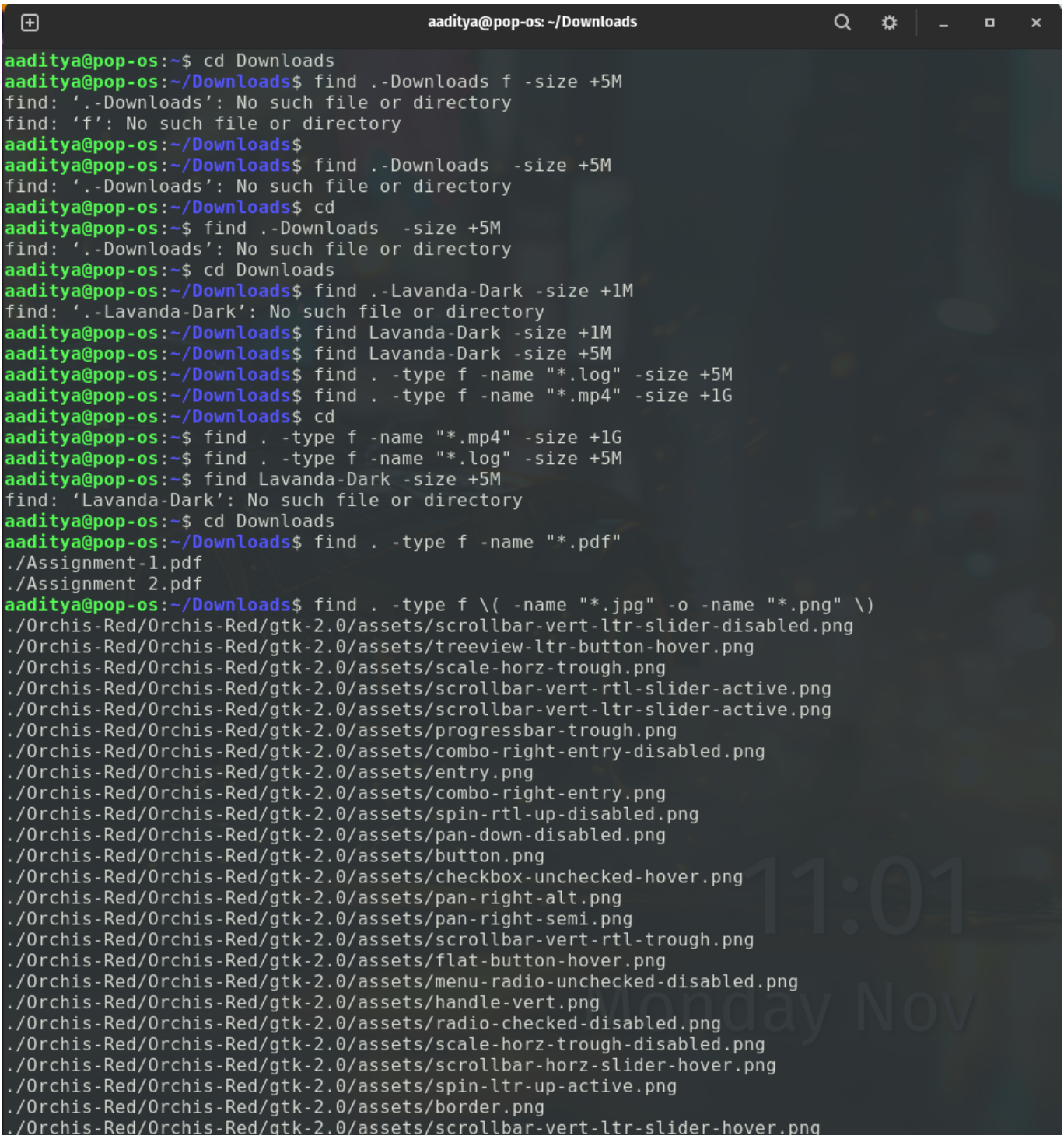
Step 2 — Use the -size flag Syntax `find <directory> -type f -size +<value><unit>`

4. Combine Extension AND Size

You can combine conditions using logical operators.

Find `.log` files larger than 5 MB `find . -type f -name "*.log" -size +5M`

Output:



Q3.Generate Fibonacci series up to a given number of terms.

Example Output: Enter limit: 8 0 1 1 2 3 5 8 13

1. Create a New Bash Script

Open a terminal and create a script file `nano fibonacci.sh`

2.Paste the Following Script

```
#!/bin/bash
```

```
# Read number of terms
read -p "Enter limit: " n

a=0
b=1

# Print Fibonacci series
for (( i=0; i<n; i++ ))
do
    echo -n "$a "
    fn=$((a + b))
    a=$b
    b=$fn
done

echo
```

3. Save and Exit

In nano: CTRL + O (save) ENTER CTRL + X (exit)

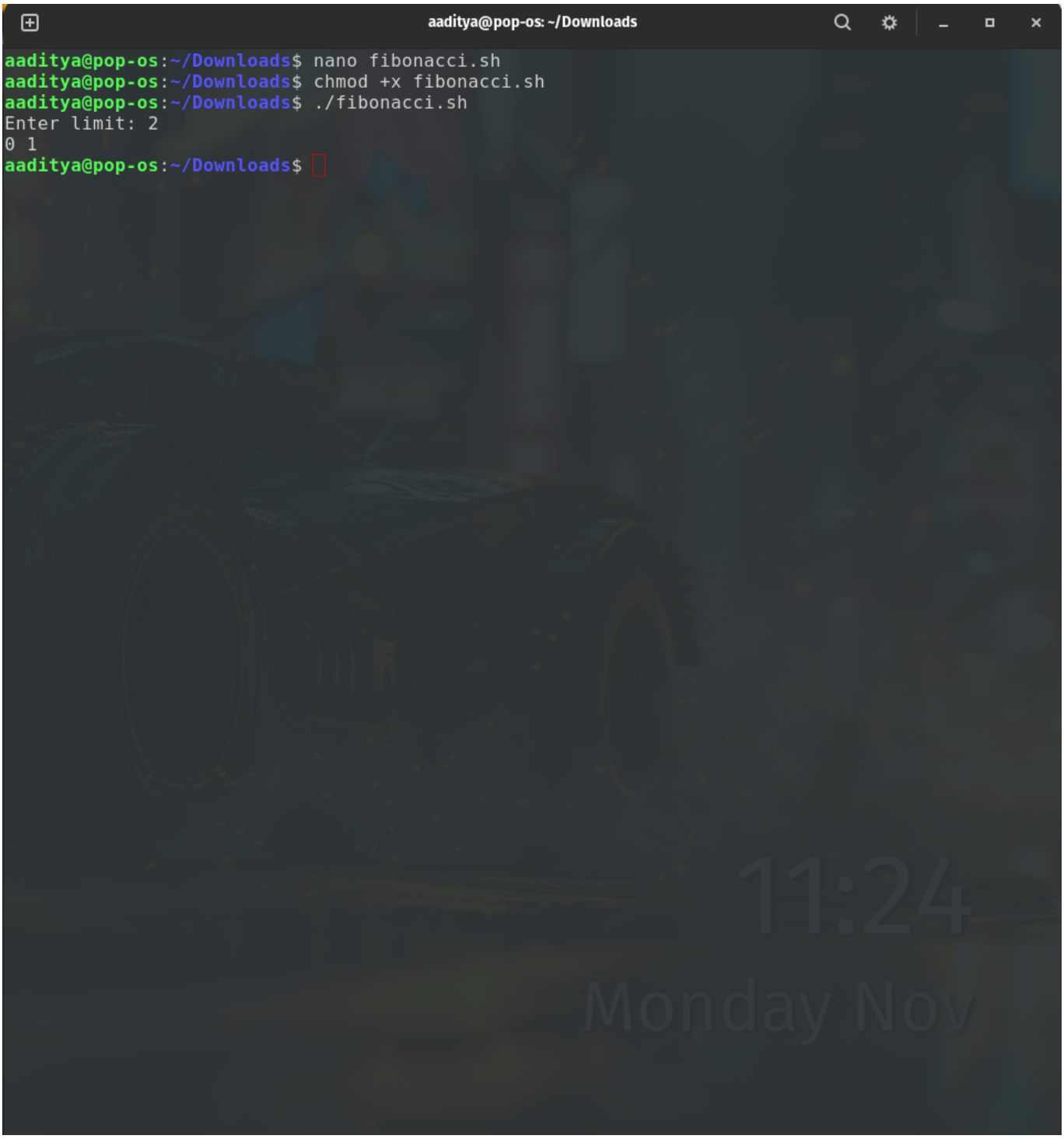
4. Make the Script Executable

```
chmod +x fibonacci.sh
```

5. Run the Script

```
./fibonacci.sh
```

Output:



Q4.Check if a file is readable, writable, or executable by the user.

1. Create a Bash Script

Open a terminal and create a new script: `nano check_permissions.sh`

2. Paste the Following Script

```
#!/bin/bash

# Ask for the file name
read -p "Enter file name: " file
```

```
# Check if file exists
if [[ ! -e "$file" ]]; then
    echo "Error: File does not exist."
    exit 1
fi

# Check readability
if [[ -r "$file" ]]; then
    echo "Readable: YES"
else
    echo "Readable: NO"
fi

# Check writability
if [[ -w "$file" ]]; then
    echo "Writable: YES"
else
    echo "Writable: NO"
fi

# Check executability
if [[ -x "$file" ]]; then
    echo "Executable: YES"
else
    echo "Executable: NO"
fi
```

3. Save and Exit

In nano: CTRL + O ENTER CTRL + X

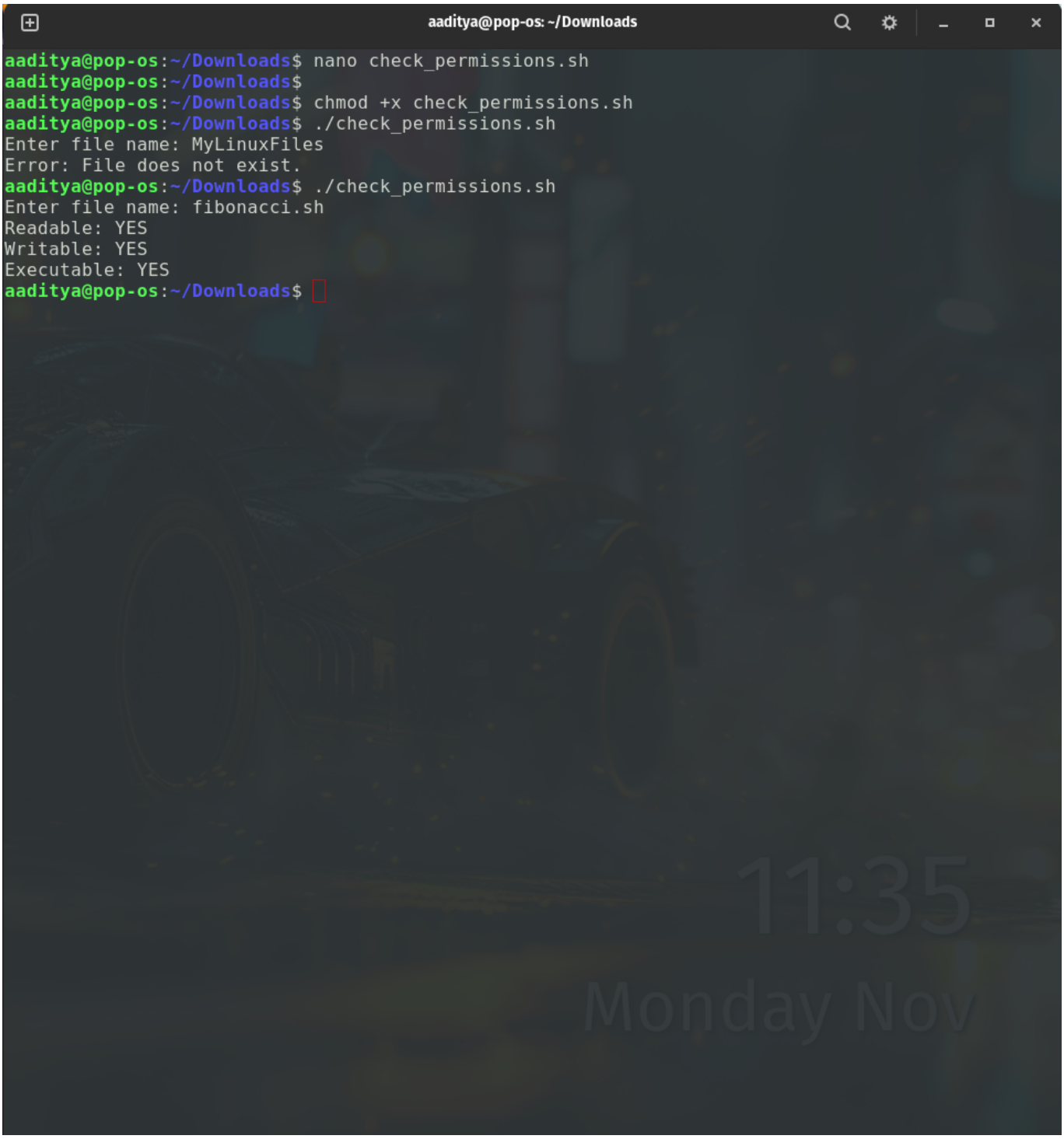
4. Make the Script Executable

```
chmod +x check_permissions.sh
```

5.5. Run the Script

```
./check_permissions.sh
```

Output:



Q5.Display system information (date, uptime, users, memory, disk usage)

1. Create the Script File

Open your terminal and create a new script: `nano system_info.sh`

2.2. Paste the Following Script

```
#!/bin/bash

echo "===== SYSTEM INFORMATION ====="
```

```
# Date and Time
echo -e "\n Date & Time:"
date

# System Uptime
echo -e "\n Uptime:"
uptime -p   # human-readable uptime

# Logged-in Users
echo -e "\n Logged-in Users:"
who

# Memory Usage
echo -e "\n Memory Usage:"
free -h

# Disk Usage
echo -e "\n Disk Usage:"
df -h

echo -e "\n===== END ====="
```

3. Save and Exit

In nano: CTRL + O ENTER CTRL + X

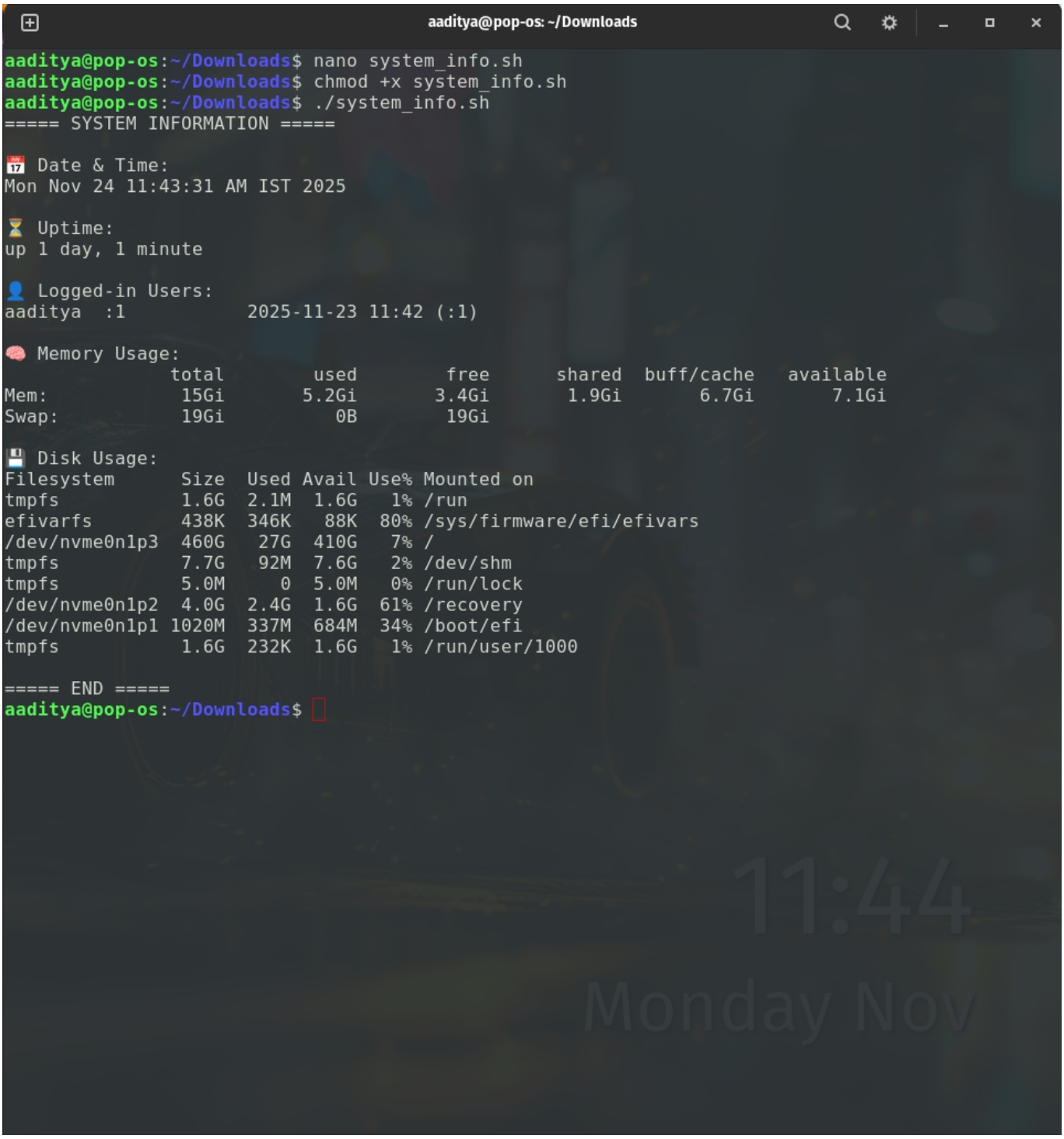
4. Make the Script Executable

```
chmod +x system_info.sh
```

5. Run the Script

```
./system_info.sh
```

Output:



Q6.Continuously monitor and log top memory-consuming processes.

1. Create a Bash Script

Open a terminal and create a script: `nano monitor_memory.sh`

2. Paste the Following Script

```
#!/bin/bash

LOGFILE="memory_log.txt"
INTERVAL=5   # seconds between checks
```

```
TOP=10       # number of top processes to log

echo "Starting memory monitor..."
echo "Logging top $TOP memory-consuming processes every $INTERVAL seconds."
echo "Log file: $LOGFILE"
echo "-----------------------------------------" >> $LOGFILE

while true
do
    echo -e "\n===== $(date) =====" >> $LOGFILE
    ps aux --sort=-%mem | head -n $((TOP + 1)) >> $LOGFILE
    echo "Logged at $(date)"
    sleep $INTERVAL
done
```

3. Save and Exit

In nano: CTRL + O ENTER CTRL + X

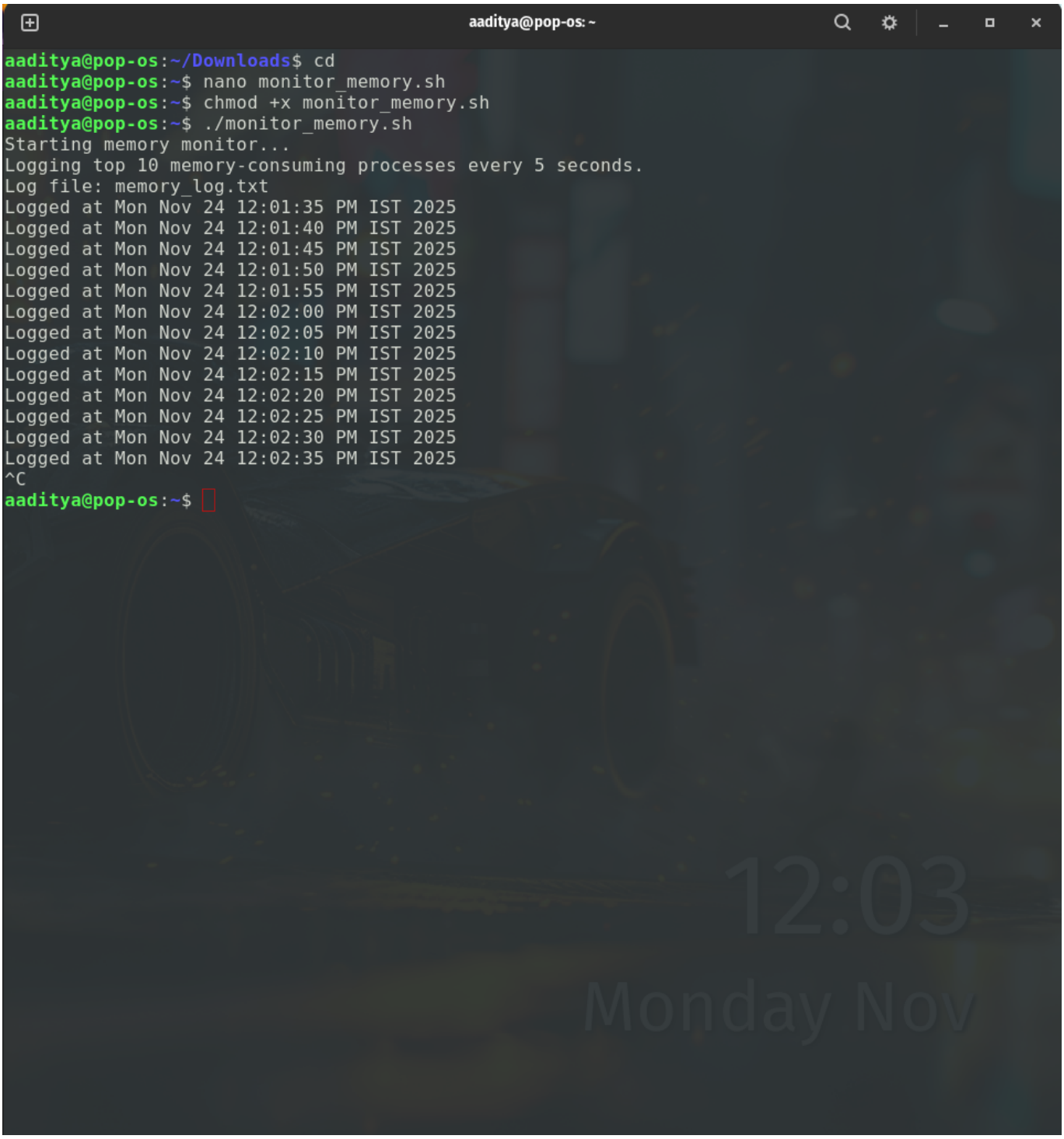
4. Make the Script Executable

```
chmod +x monitor_memory.sh
```

5. Run the Script:

```
./monitor_memory.sh
```

Output:



Q7.Take a filename as input and display the number of lines, words, and characters.

1. Create the Script File

Open your terminal and create a script: `nano file_stats.sh`

2. Paste the Following Script

```
#!/bin/bash

# Ask for filename
read -p "Enter filename: " file
```

```
# Check if file exists
if [[ ! -f "$file" ]]; then
    echo "Error: File does not exist."
    exit 1
fi

# Count lines, words, and characters using wc
lines=$(wc -l < "$file")
words=$(wc -w < "$file")
chars=$(wc -m < "$file")

echo "File: $file"
echo "Lines: $lines"
echo "Words: $words"
echo "Characters: $chars"
```

3. Save and Exit

In nano: CTRL + O ENTER CTRL + X

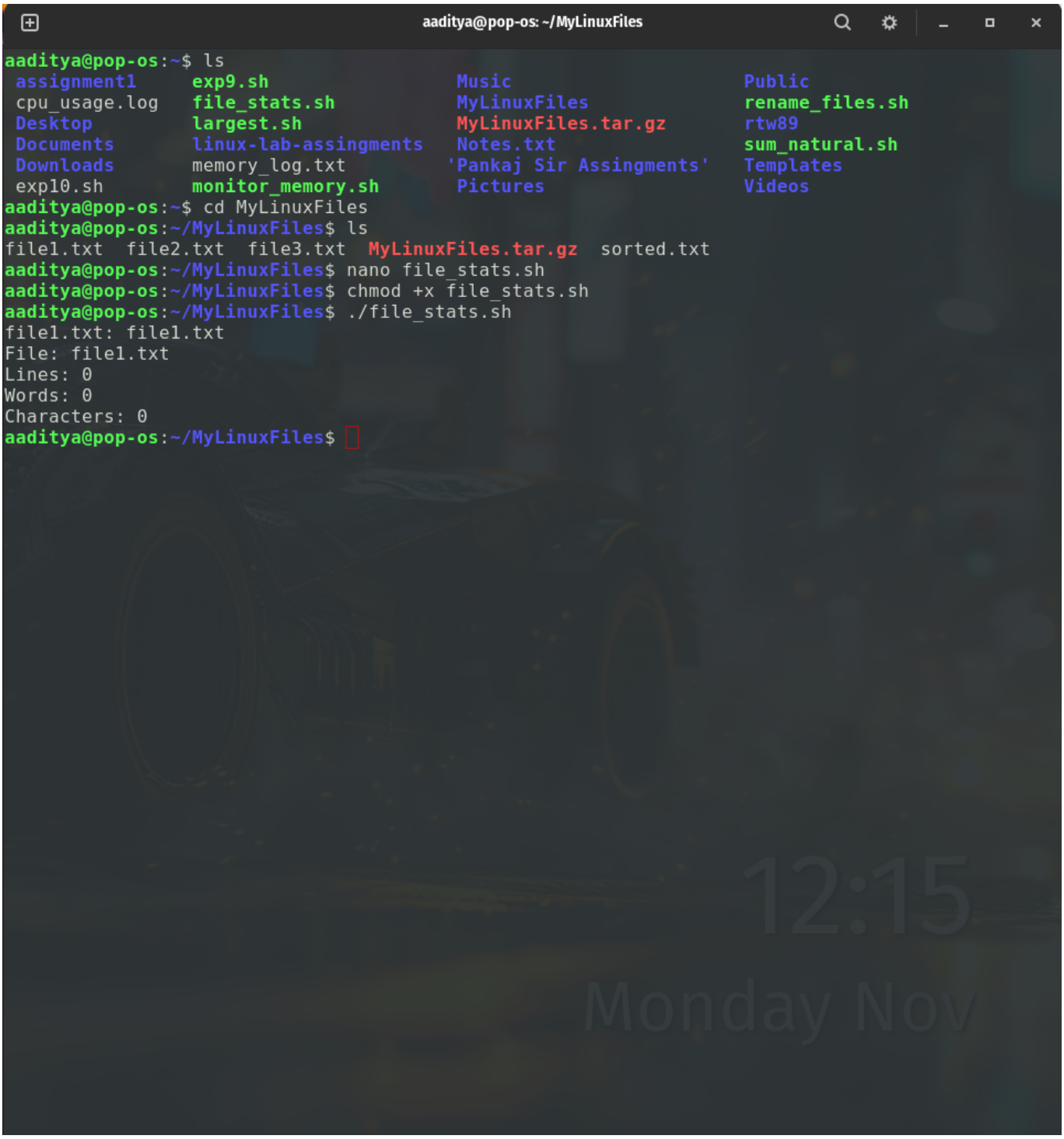
4. Make the Script Executable

```
chmod +x file_stats.sh
```

5. Run the Script

```
./file_stats.sh
```

Output:



Q8.Accept multiple numbers and sort them in ascending order.

1. Create a New Script

Open your terminal: `nano sort_numbers.sh`

2. Paste the Following Bash Script

```
#!/bin/bash

# Ask user for numbers
read -p "Enter numbers separated by spaces: " -a nums
```

```
# Sort the numbers
sorted=$(printf "%s\n" "${nums[@]}" | sort -n)

echo "Sorted numbers (ascending):"
echo "$sorted"
```

3. Save and Exit

In nano: CTRL + O ENTER CTRL + X

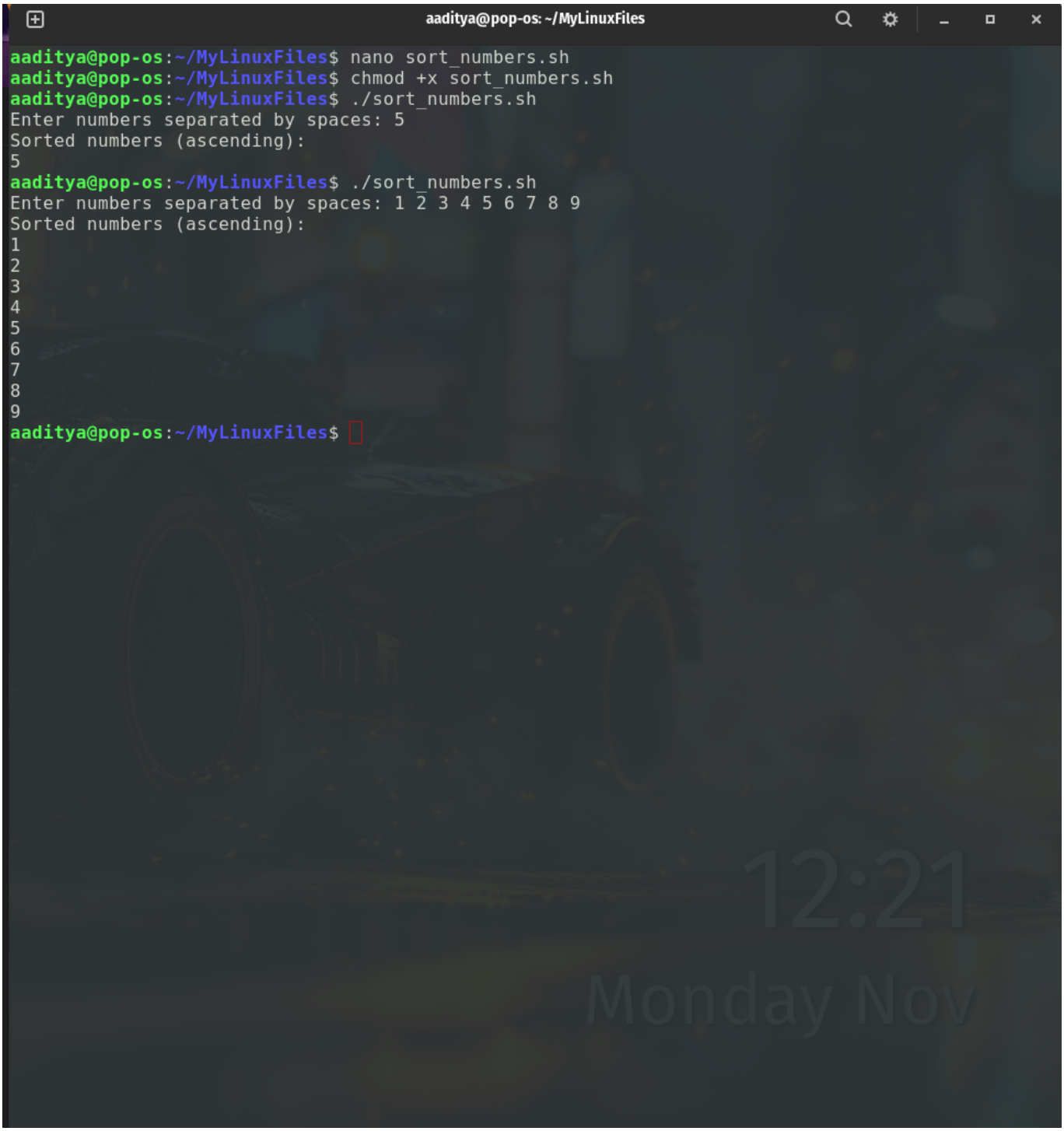
4. Make the Script Executable

```
chmod +x sort_numbers.sh
```

5. Run the Script

```
./sort_numbers.sh
```

Output:



Q9.Calculate the GCD and LCM of two given numbers.

1. Create a New Script

Open your terminal: `nano gcd_lcm.sh`

2. Paste the Following Bash Script

```
#!/bin/bash

# Ask for two numbers
read -p "Enter first number: " a
read -p "Enter second number: " b
```

```
# Validate input
if ! [[ "$a" =~ ^[0-9]+$ ]] || ! [[ "$b" =~ ^[0-9]+$ ]]; then
    echo "Error: Please enter valid positive integers."
    exit 1
fi

# Function to calculate GCD using Euclidean algorithm
gcd() {
    x=$1
    y=$2
    while [ $y -ne 0 ]; do
        r=$((x % y))
        x=$y
        y=$r
    done
    echo $x
}

# Function to calculate LCM
lcm() {
    x=$1
    y=$2
    # LCM = (a * b) / GCD
    echo $(( (x * y) / $(gcd $x $y) ))
}

GCD=$(gcd $a $b)
LCM=$(lcm $a $b)

echo "GCD of $a and $b is: $GCD"
echo "LCM of $a and $b is: $LCM"
```

3. Save and Exit

In nano: CTRL + O ENTER CTRL + X

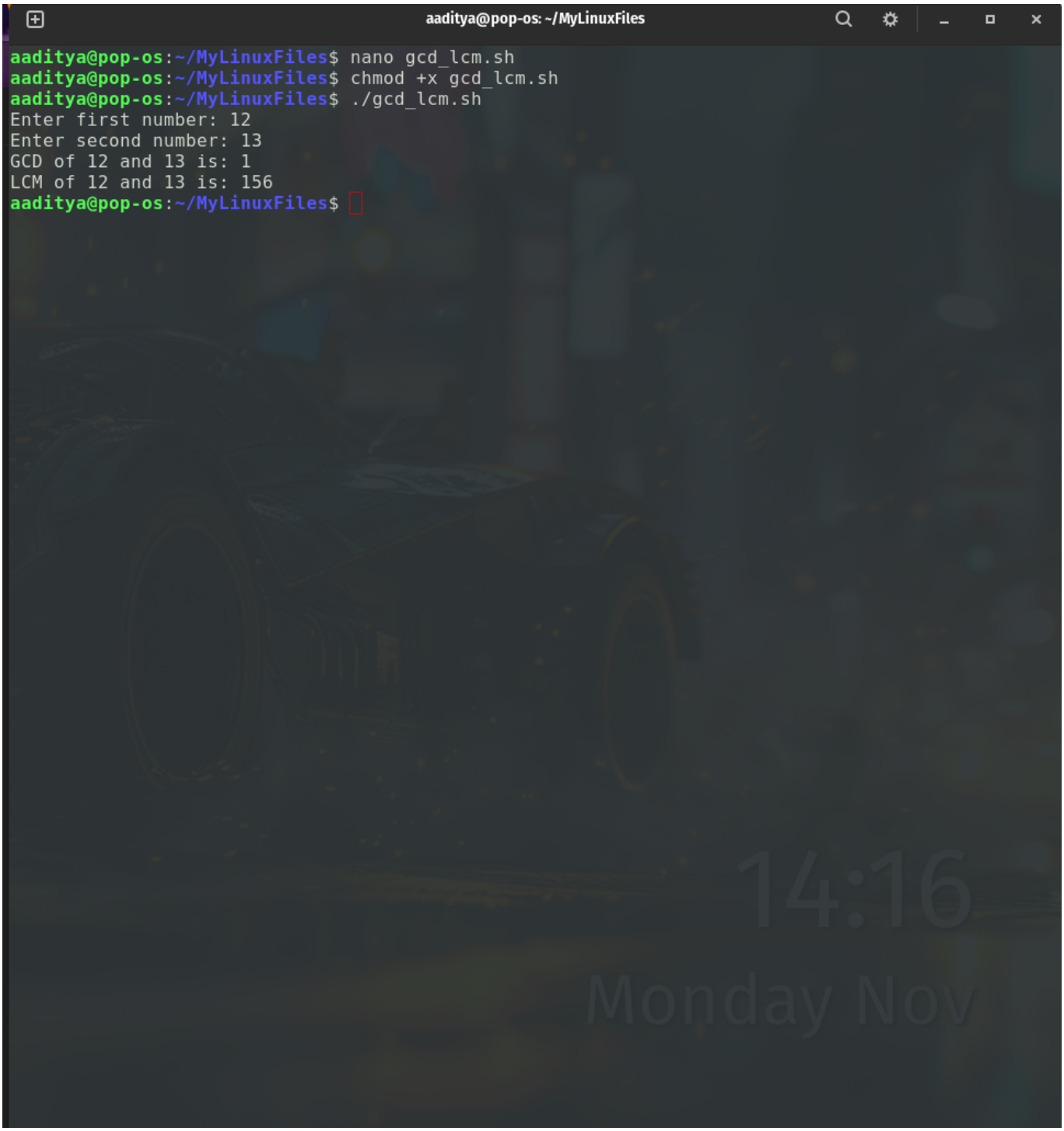
4. Make the Script Executable

```
chmod +x gcd_lcm.sh
```

5. Run the Script

```
./gcd_lcm.sh
```

Output:



Q10.Check whether an entered string is a palindrome or not.

1. Create a New Script

Open your terminal: `nano palindrome_check.sh`

2. Paste the Following Bash Script

```
#!/bin/bash

# Ask for input string
read -p "Enter a string: " str
```

```
# Remove spaces and convert to lowercase for uniformity
clean_str=$(echo "$str" | tr -d ' ' | tr '[:upper:]' '[:lower:]')

# Reverse the string
rev_str=$(echo "$clean_str" | rev)

# Check if palindrome
if [ "$clean_str" == "$rev_str" ]; then
    echo "\"$str\" is a palindrome."
else
    echo "\"$str\" is not a palindrome."
fi
```

3. Save and Exit

In nano: CTRL + O ENTER CTRL + X

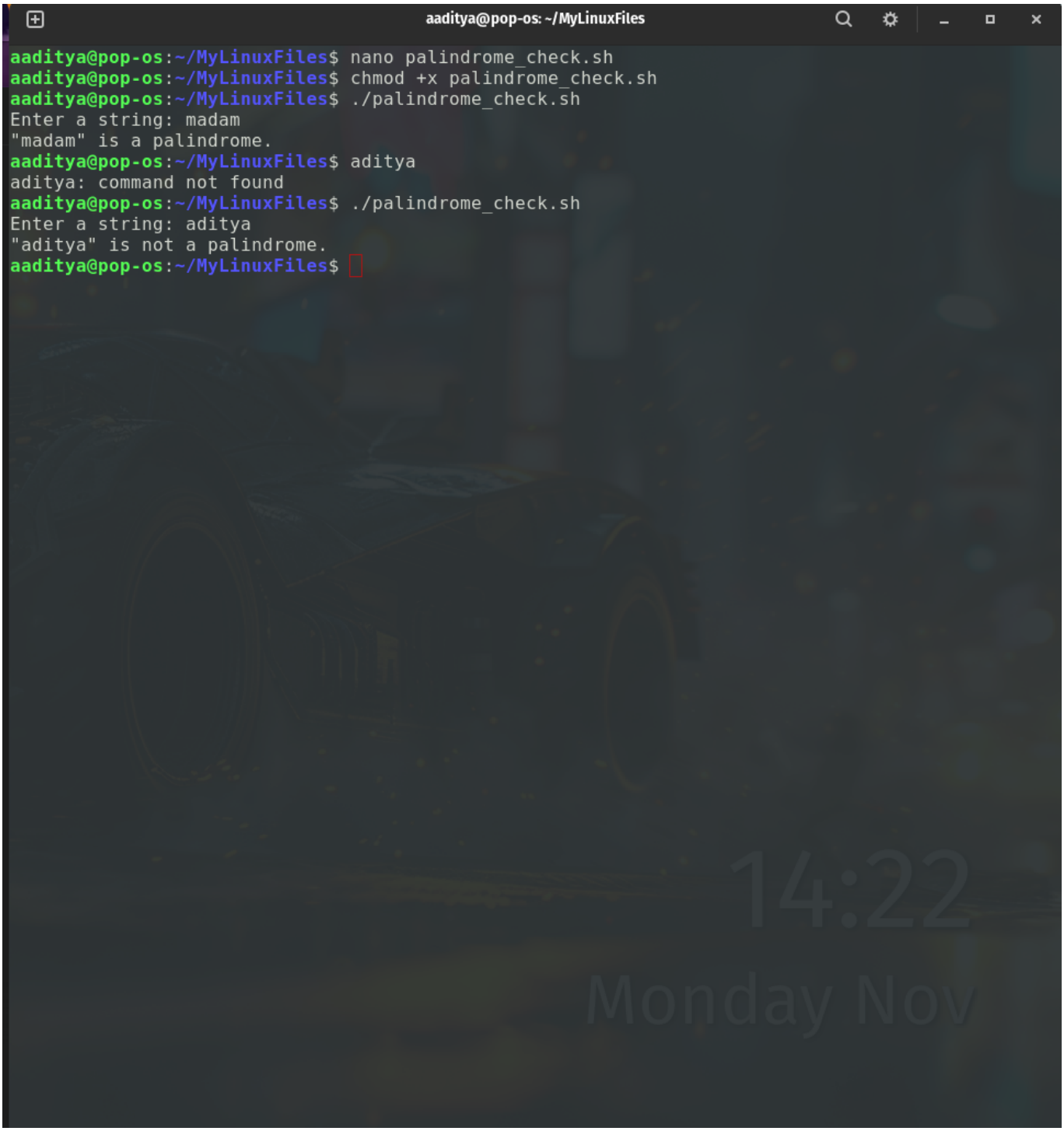
4. Make the Script Executable

```
chmod +x palindrome_check.sh
```

5. Run the Script

```
./palindrome_check.sh
```

Output:



Q11. Calculate and display the length of a string

1. Create a New Script

Open your terminal: `nano string_length.sh`

2. Paste the Following Bash Script

```
#!/bin/bash

# Ask for input string
read -p "Enter a string: " str
```

```
# Calculate length
length=${#str}

# Display result
echo "The length of the string is: $length"
```

3. Save and Exit

In nano: CTRL + O ENTER CTRL + X

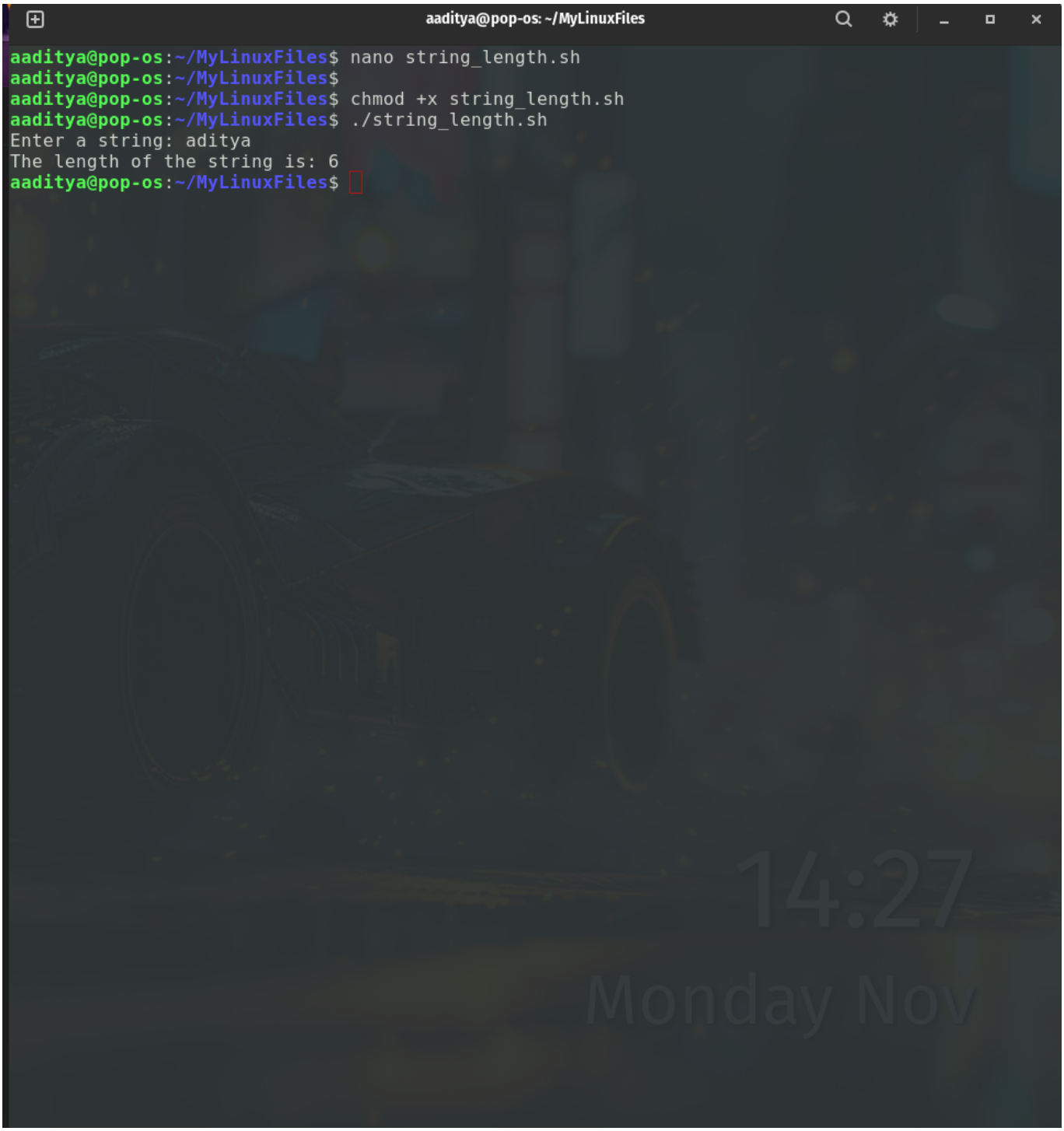
4. Make the Script Executable

```
chmod +x string_length.sh
```

5. Run the Script

```
./string_length.sh
```

Output:



Q12. Reverse a given string

1. Create a New Script

Open your terminal: `nano reverse_string.sh`

2. Paste the Following Bash Script

```
#!/bin/bash

# Ask for input string
read -p "Enter a string: " str
```

```
# Reverse the string
rev_str=$(echo "$str" | rev)

# Display the reversed string
echo "Reversed string: $rev_str"
```

3. Save and Exit

In nano: CTRL + O ENTER CTRL + X

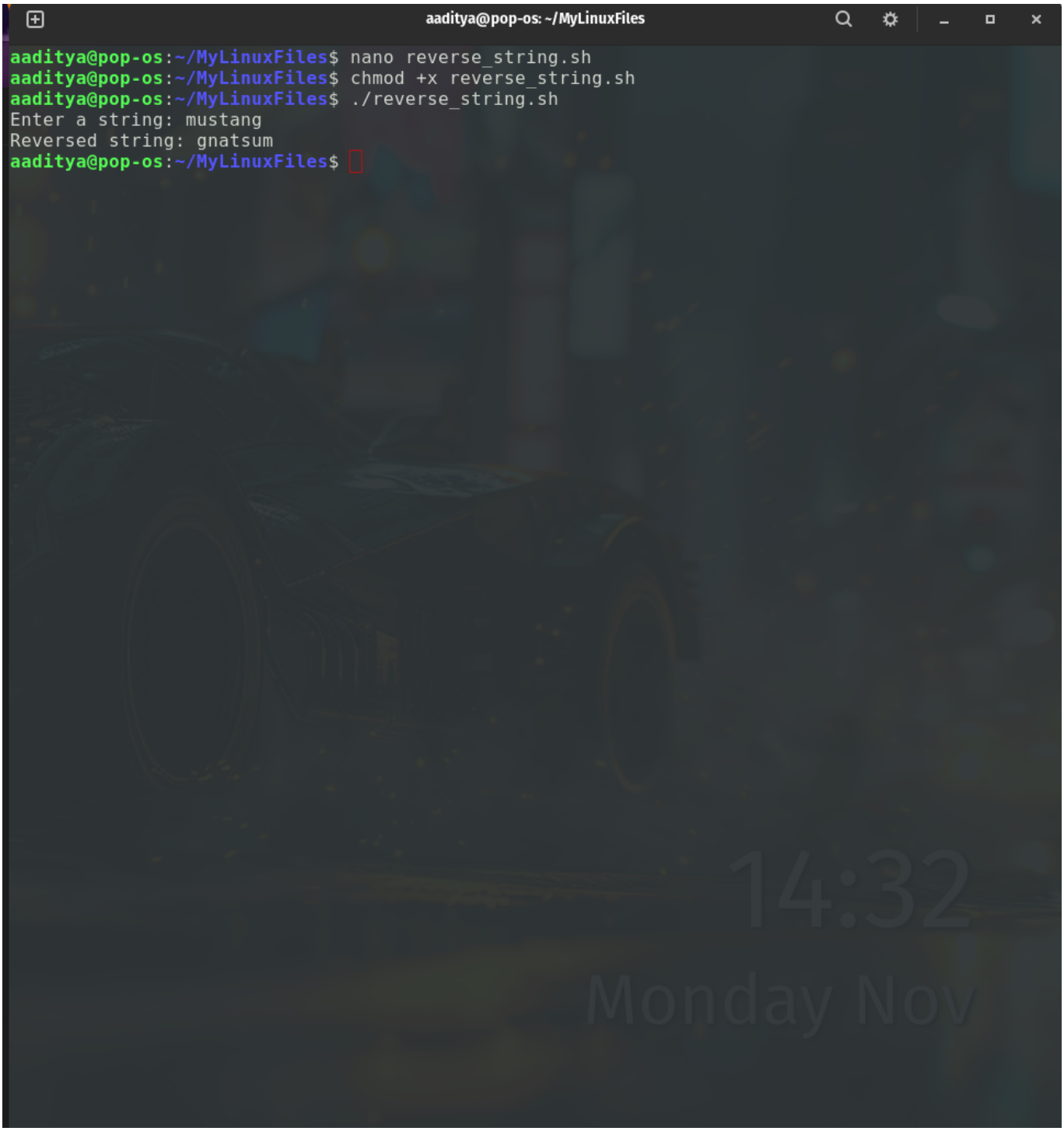
4. Make the Script Executable

```
chmod +x reverse_string.sh
```

5. Run the Script

```
./reverse_string.sh
```

Output:



Q13.Concatenate two input strings.

1. Create a New Script

Open your terminal: `nano concatenate_strings.sh`

2. Paste the Following Bash Script

```
#!/bin/bash

# Ask for first string
read -p "Enter first string: " str1
```

```
# Ask for second string
read -p "Enter second string: " str2

# Concatenate strings
concat="$str1$str2"

# Display result
echo "Concatenated string: $concat"
```

3. Save and Exit

In nano: CTRL + O ENTER CTRL + X

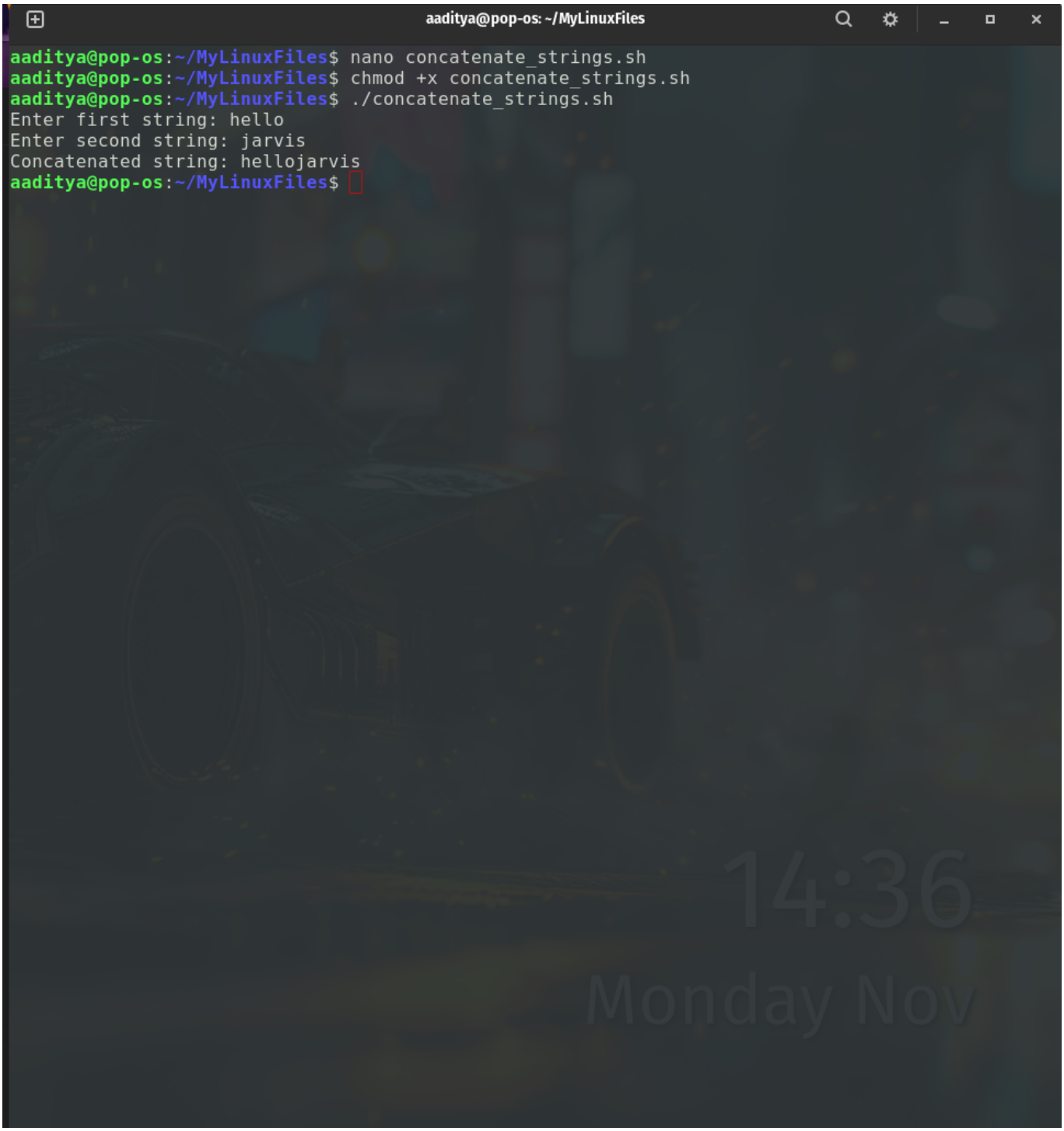
4. Make the Script Executable

```
chmod +x concatenate_strings.sh
```

5. Run the Script

```
./concatenate_strings.sh
```

Output:


aaditya@pop-os: ~/MyLinuxFiles
aaditya@pop-os:~/MyLinuxFiles$ nano concatenate_strings.sh
aaditya@pop-os:~/MyLinuxFiles$ chmod +x concatenate_strings.sh
aaditya@pop-os:~/MyLinuxFiles$ ./concatenate_strings.sh
Enter first string: hello
Enter second string: jarvis
Concatenated string: hellojarvis
aaditya@pop-os:~/MyLinuxFiles$
14:36
Monday Nov
