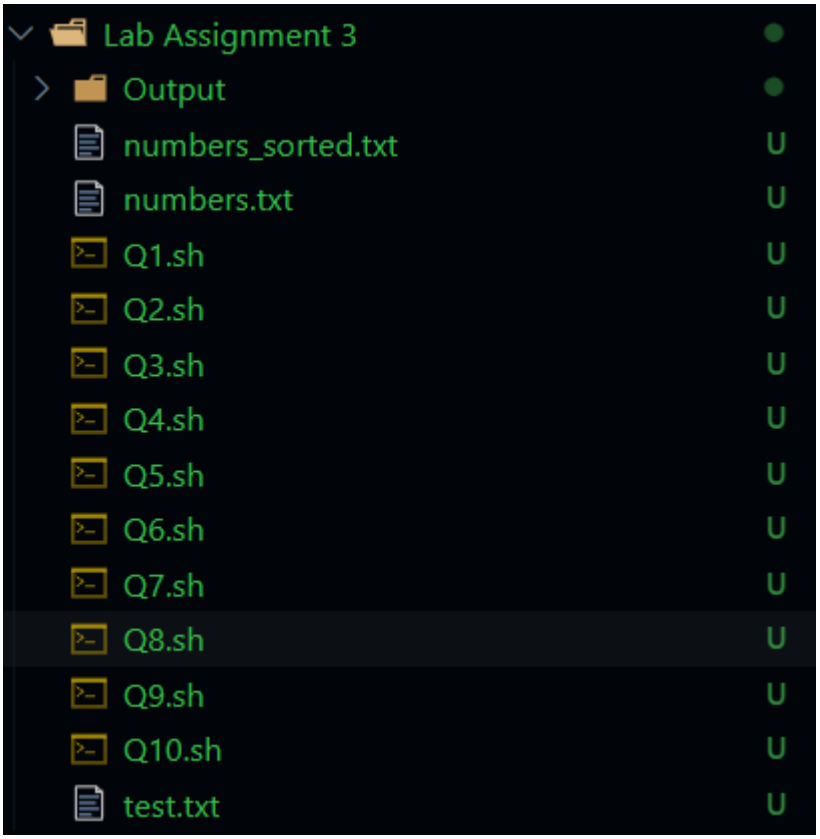
Lab Assignment 3

Name: Manobal Singh Bagady
SID: 21104129
Semester: 8
Branch: Electrical Engineering
Due Date: Feb 14, 2025

The folder structure is as follows:

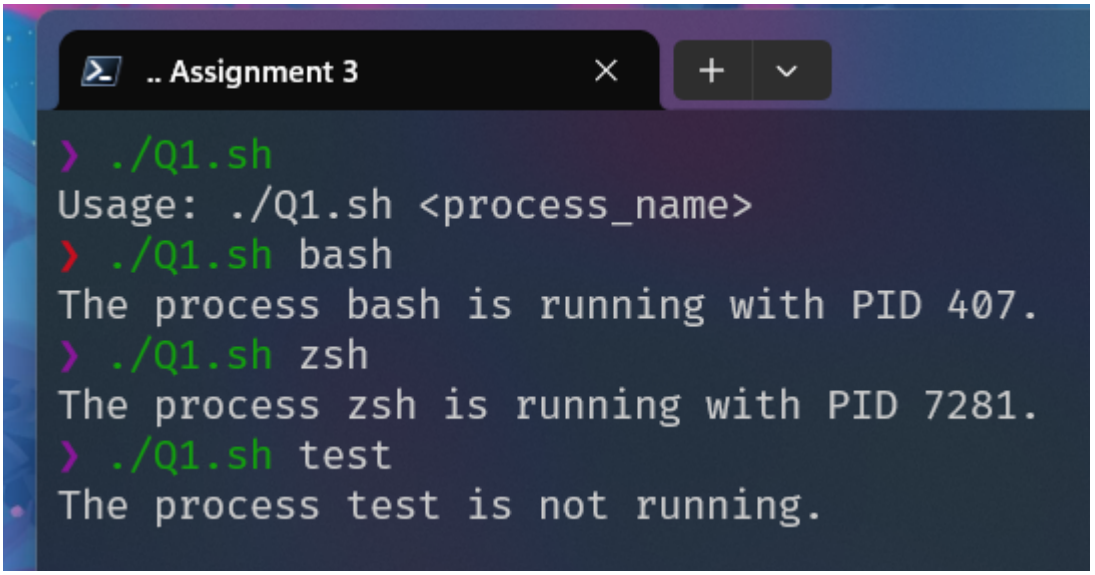


Q1.)

Code:

```bash
#!/bin/bash

# Question 1
# Script to check if a process is running and display it's PID.

# Check if the number of arguments is less than 1
if [ $# -lt 1 ]
then
    echo "Usage: $0 <process_name>"
    exit 1
fi

# Get the process name
process_name=$1

# Get the PID of the process
pid=$(pgrep -n $process_name)

# Check if the process is running
if [ -z "$pid" ]
then
    echo "The process $process_name is not running."
else
    echo "The process $process_name is running with PID $pid."
fi
```

Output:

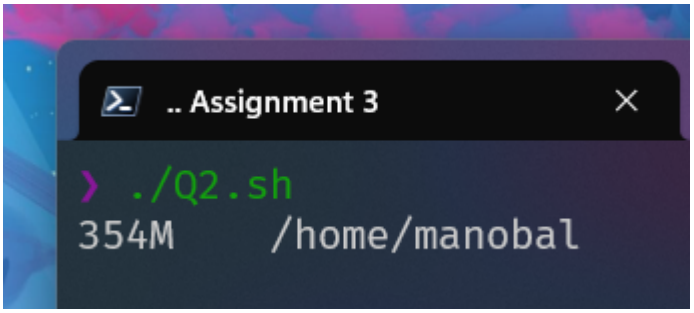


Q2.)

Code:

```
#!/bin/bash

# Question 2
# script to display disk usage of home directory and display it in
human readable format.

# Display disk usage of home directory
du -sh ~
```

Output:

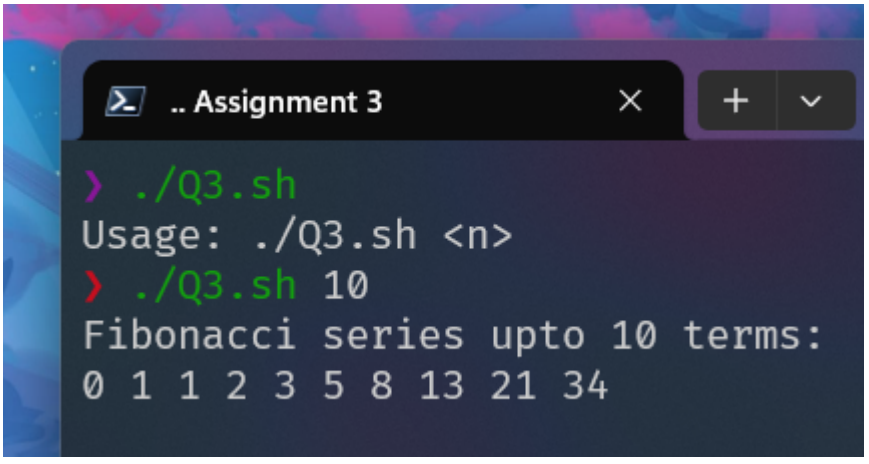


Q3.)

Code:

```bash
#!/bin/bash

# Question 3
# Fibonacci series

# Check if the number of arguments is less than 1
if [ $# -lt 1 ]; then
    echo "Usage: $0 <n>"
    exit 1
fi

# Get the number of terms
n=$1

# Adding the first two numbers to the array
fibonacci=(0 1)

# loop to add the next numbers to the array
for ((i = 2; i < n; i++)); do
    # adding the next number to the array
    fibonacci[i]=$((fibonacci[i - 1] + fibonacci[i - 2]))
done

# Display the fibonacci series
echo "Fibonacci series upto $n terms:"
echo "${fibonacci[@]}"
```

Output:

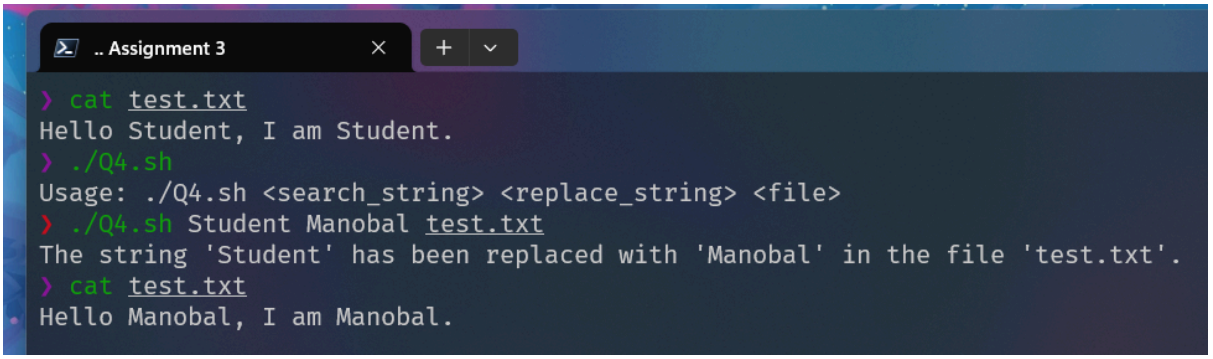


Q4.)

Code:

```
#!/bin/bash

# Question 4
# Script to search and replace a string in a file.

# Check if the number of arguments is less than 3
if [ $# -lt 3 ]; then
    echo "Usage: $0 <search_string> <replace_string> <file>"
    exit 1
fi

# Get the search string, replace string and file
search_string=$1
replace_string=$2
file=$3

# Search and replace the string in the file
sed -i "s/$search_string/$replace_string/g" $file

echo "The string '$search_string' has been replaced with
'$replace_string' in the file '$file'."
```

Output:

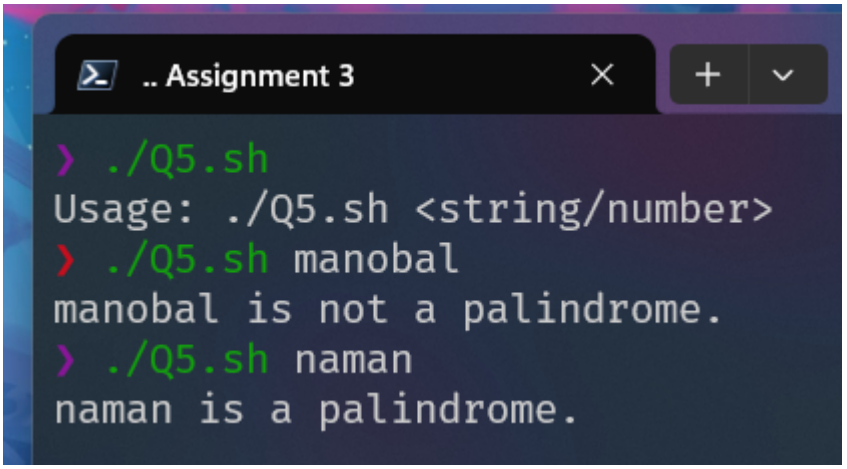


Q5.)

Code:

```bash
#!/bin/bash

# Question 5
# Check given string or number is palindrome or not

# Check if the number of arguments is less than 1
if [ $# -lt 1 ]; then
    echo "Usage: $0 <string/number>"
    exit 1
fi

# Get the string or number
input=$1

# Reverse the input
reverse=$(echo $input | rev)

# Check if the input is a palindrome
if [ "$input" == "$reverse" ]; then
    echo "$input is a palindrome."
else
    echo "$input is not a palindrome."
fi
```

Output:



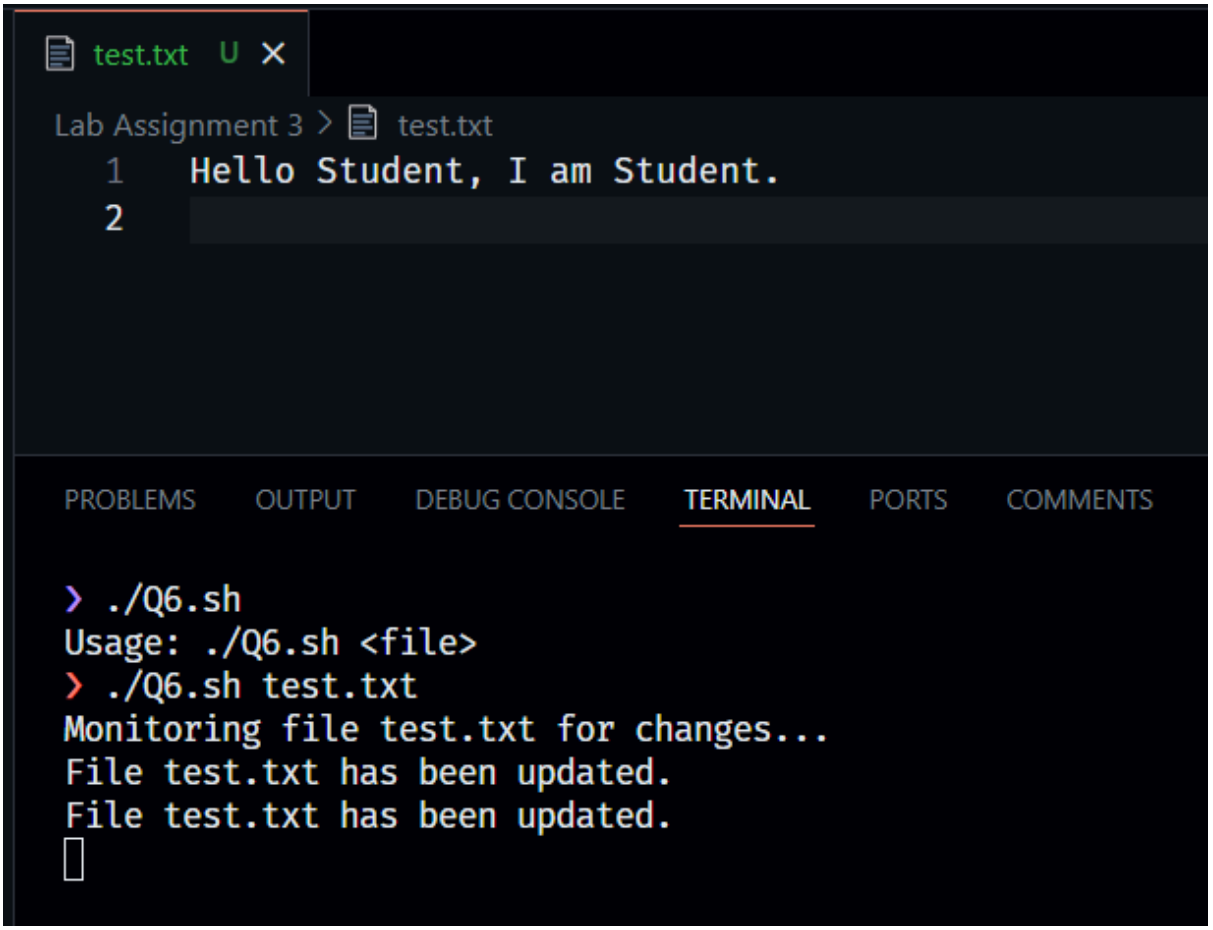
Q6.)

Code:

```
#!/bin/bash

# Question 6
# Script that monitors file for changes and displays a message when the
file is updated.

# Check if the number of arguments is less than 1
if [ $# -lt 1 ]; then
    echo "Usage: $0 <file>"
    exit 1
fi

# Get the file
file=$1

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

# Get the initial timestamp of the file
timestamp=$(stat -c %Y $file)

echo "Monitoring file $file for changes..."

# Monitor the file for changes
while true; do
    new_timestamp=$(stat -c %Y $file)

    if [ $new_timestamp -gt $timestamp ]; then
        echo "File $file has been updated."
        timestamp=$new_timestamp
    fi

    sleep 1
done
```

Output:


test.txt U
Lab Assignment 3 > test.txt
1 Hello Manobal, I am Manobal.
2
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS
> ./Q6.sh
Usage: ./Q6.sh <file>
> ./Q6.sh test.txt
Monitoring file test.txt for changes...

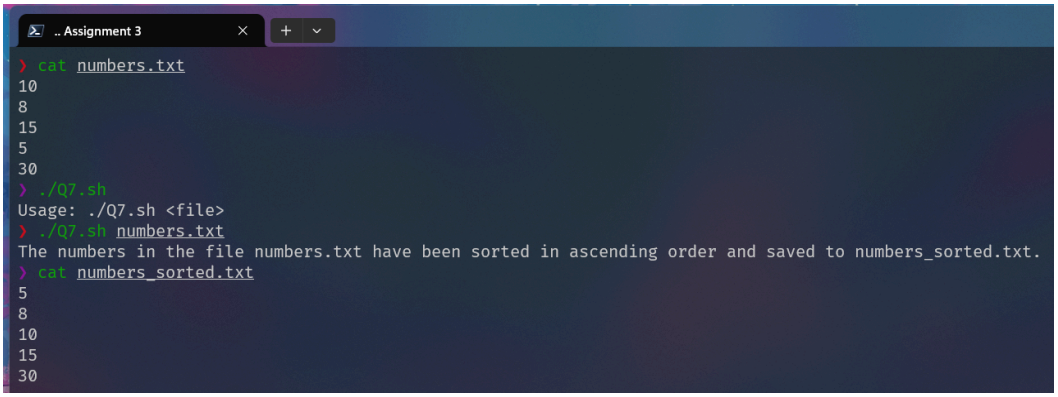

test.txt U
Lab Assignment 3 > test.txt
1 Hello Student, I am Student.
2
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS
> ./Q6.sh
Usage: ./Q6.sh <file>
> ./Q6.sh test.txt
Monitoring file test.txt for changes...
File test.txt has been updated.
File test.txt has been updated.

Q7.)

Code:

```bash
#!/bin/bash

# Question 7
# Script to read numbers from a file, sort them in ascending order and save them to a new file.

# Check if the number of arguments is less than 1
if [ $# -lt 1 ]; then
    echo "Usage: $0 <file>"
    exit 1
fi

# Get the file
file=$1

# Check if the file exists
if [ ! -f $file ]; then
    echo "The file $file does not exist."
    exit 1
fi

new_file="${file%.*}_sorted.${file##*.}"

sort -n "$file" >"$new_file"

echo "The numbers in the file $file have been sorted in ascending order and saved to $new_file."
```

Output:

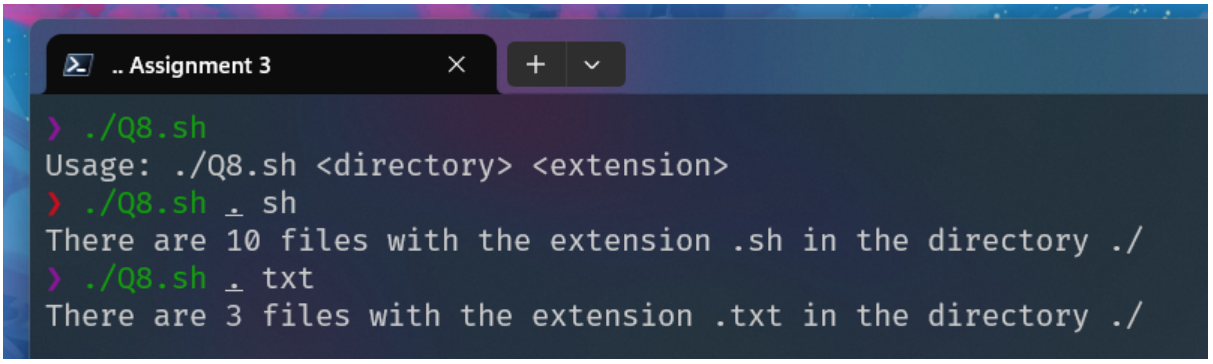


Q8.)

Code:

```bash
#!/bin/bash

# Question 8
# Script to count files with a specific extension in a directory.

# Check if the number of arguments is less than 2
if [ $# -lt 2 ]; then
    echo "Usage: $0 <directory> <extension>"
    exit 1
fi

# Get the directory and extension
directory=$1
extension=$2

# Check if the directory exists
if [ ! -d "$directory" ]; then
    echo "The directory $directory does not exist."
    exit 1
fi

# Count the number of files with the specified extension
count=$(ls -1 $directory/*.$extension 2>/dev/null | wc -l)

echo "There are $count files with the extension .$extension in the
directory $directory/"
```

Output:

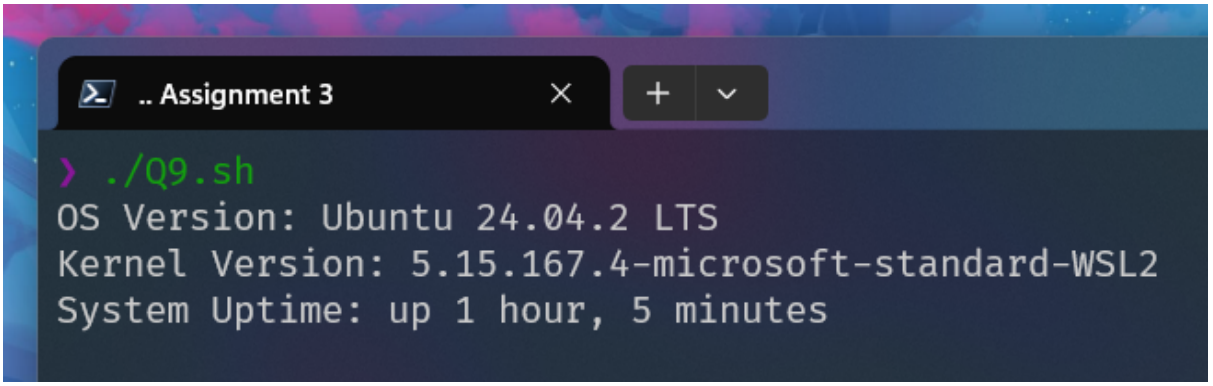


Q9.)

Code:

```bash
#!/bin/bash

# Question 9
# Script to display system information such as OS version, kernel
version, and system uptime.

# Display the OS version
os_version=$(lsb_release -d | cut -f2)
echo "OS Version: $os_version"

# Display the kernel version
kernel_version=$(uname -r)
echo "Kernel Version: $kernel_version"

# Display the system uptime
uptime=$(uptime -p)
echo "System Uptime: $uptime"
```

Output:



Q10.)

Code:

```bash
#!/bin/bash

# Question 10
# Script to display multiplication table of a number.

# Check if the number of arguments is less than 1
if [ $# -lt 1 ]; then
    echo "Usage: $0 <number>"
    exit 1
fi

# Get the number
number=$1

echo "Multiplication Table of $number:"
# Display the multiplication table of the number
for i in {1..10}; do
    result=$((number * i))
    echo "$number x $i = $result"
done
```

Output: