

Lab Assignment 3

Name: Manobal Singh Bagady

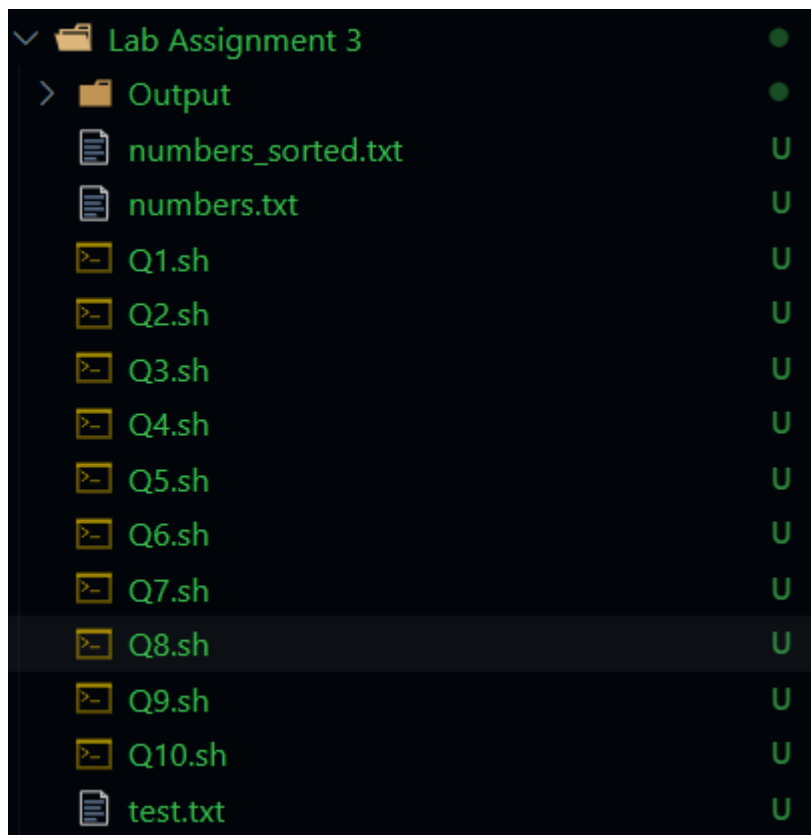
SID: 21104129

Semester: 8

Branch: Electrical Engineering

Due Date: Feb 14, 2025

The folder structure is as follows:



Q1.)

Code:

```
#!/bin/bash

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

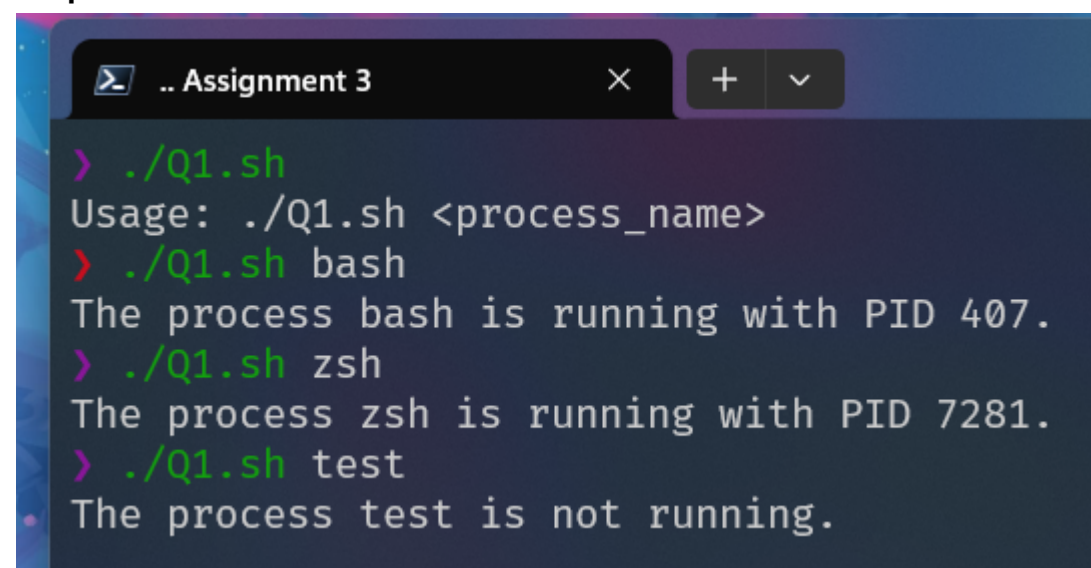
# Check if the number of arguments is less than 1
if [ $# -lt 1 ]
then
    echo "Usage: $0 <process_name>"
    exit 1
fi

# 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:



The image shows a terminal window titled "Assignment 3". The user has executed the script `./Q1.sh` with three different arguments: `bash`, `zsh`, and `test`. The output shows that `bash` and `zsh` are running with PIDs 407 and 7281 respectively, while `test` is not running.

```
> ./Q1.sh
Usage: ./Q1.sh <process_name>
> ./Q1.sh bash
The process bash is running with PID 407.
> ./Q1.sh zsh
The process zsh is running with PID 7281.
> ./Q1.sh test
The process test is not running.
```

Q2.)

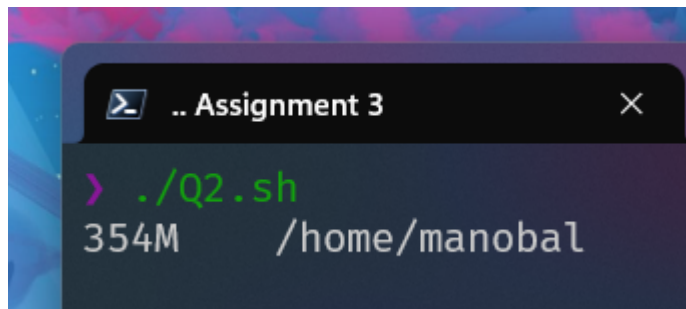
Code:

```
#!/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:

A terminal window titled '.. Assignment 3' with a close button. The prompt is '>'. The command './Q2.sh' is entered in green. The output '354M /home/manoba1' is displayed in white.

```
> ./Q2.sh
354M /home/manoba1
```

Q3.)

Code:

```
#!/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

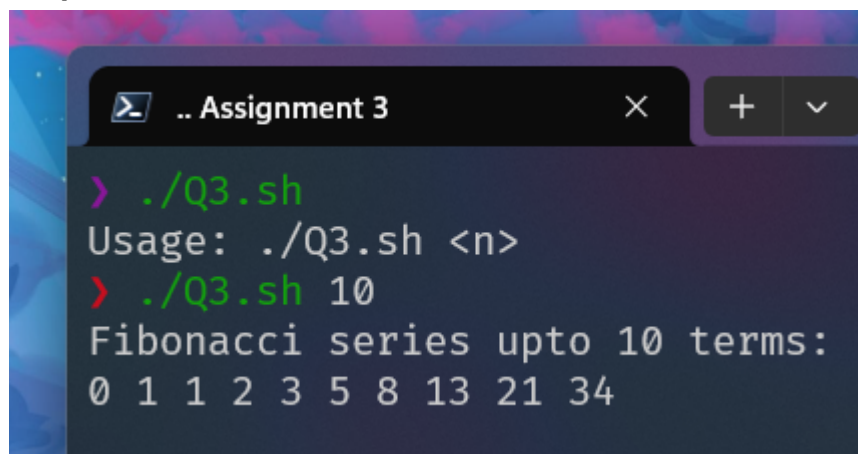
# Get the number of terms
n=$1

# 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:

A terminal window titled '.. Assignment 3' with standard window controls (close, maximize, and a dropdown menu). The terminal shows the execution of a script named Q3.sh. The first command is './Q3.sh', which results in the output 'Usage: ./Q3.sh <n>'. The second command is './Q3.sh 10', which results in the output 'Fibonacci series upto 10 terms:' followed by the sequence of numbers '0 1 1 2 3 5 8 13 21 34' on the next line.

```
> ./Q3.sh
Usage: ./Q3.sh <n>
> ./Q3.sh 10
Fibonacci series upto 10 terms:
0 1 1 2 3 5 8 13 21 34
```

Q4.)

Code:

```
#!/bin/bash

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

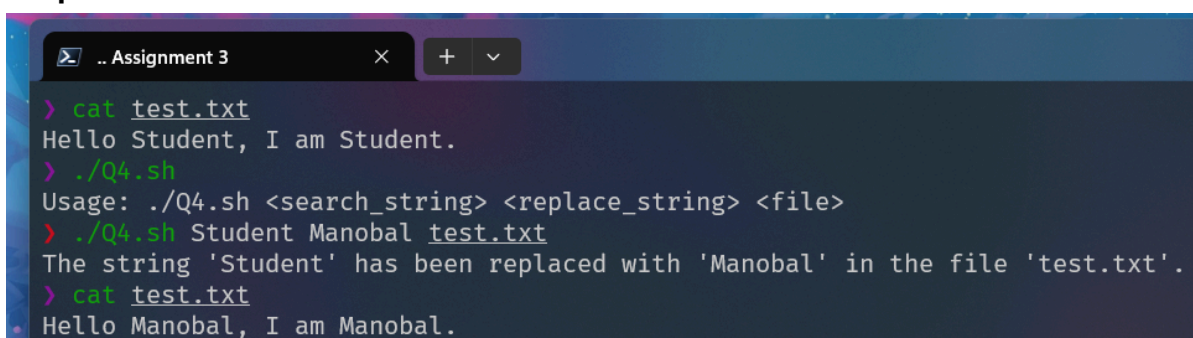
# Check if the number of arguments is less than 3
if [ $# -lt 3 ]; then
    echo "Usage: $0 <search_string> <replace_string> <file>"
    exit 1
fi

# 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:

A terminal window titled '.. Assignment 3' with standard window controls. It shows a sequence of commands and their outputs. First, 'cat test.txt' displays 'Hello Student, I am Student.'. Then, './Q4.sh' shows the usage message. Next, './Q4.sh Student Manobal test.txt' shows the replacement message. Finally, 'cat test.txt' shows the updated content 'Hello Manobal, I am Manobal.'.

```
> cat test.txt
Hello Student, I am Student.
> ./Q4.sh
Usage: ./Q4.sh <search_string> <replace_string> <file>
> ./Q4.sh Student Manobal test.txt
The string 'Student' has been replaced with 'Manobal' in the file 'test.txt'.
> cat test.txt
Hello Manobal, I am Manobal.
```

Q5.)

Code:

```
#!/bin/bash

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

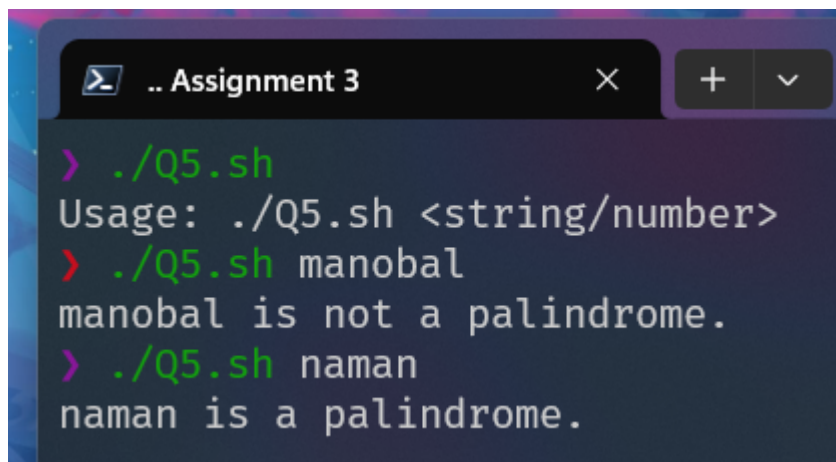
# Check if the number of arguments is less than 1
if [ $# -lt 1 ]; then
    echo "Usage: $0 <string/number>"
    exit 1
fi

# 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:



The image shows a terminal window titled '.. Assignment 3'. The user runs the script `./Q5.sh` without arguments, which displays the usage message: `Usage: ./Q5.sh <string/number>`. Then, the user runs `./Q5.sh manobal`, and the script outputs `manobal is not a palindrome.`. Finally, the user runs `./Q5.sh naman`, and the script outputs `naman is a palindrome.`

```
> ./Q5.sh
Usage: ./Q5.sh <string/number>
> ./Q5.sh manobal
manobal is not a palindrome.
> ./Q5.sh naman
naman is a palindrome.
```

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)

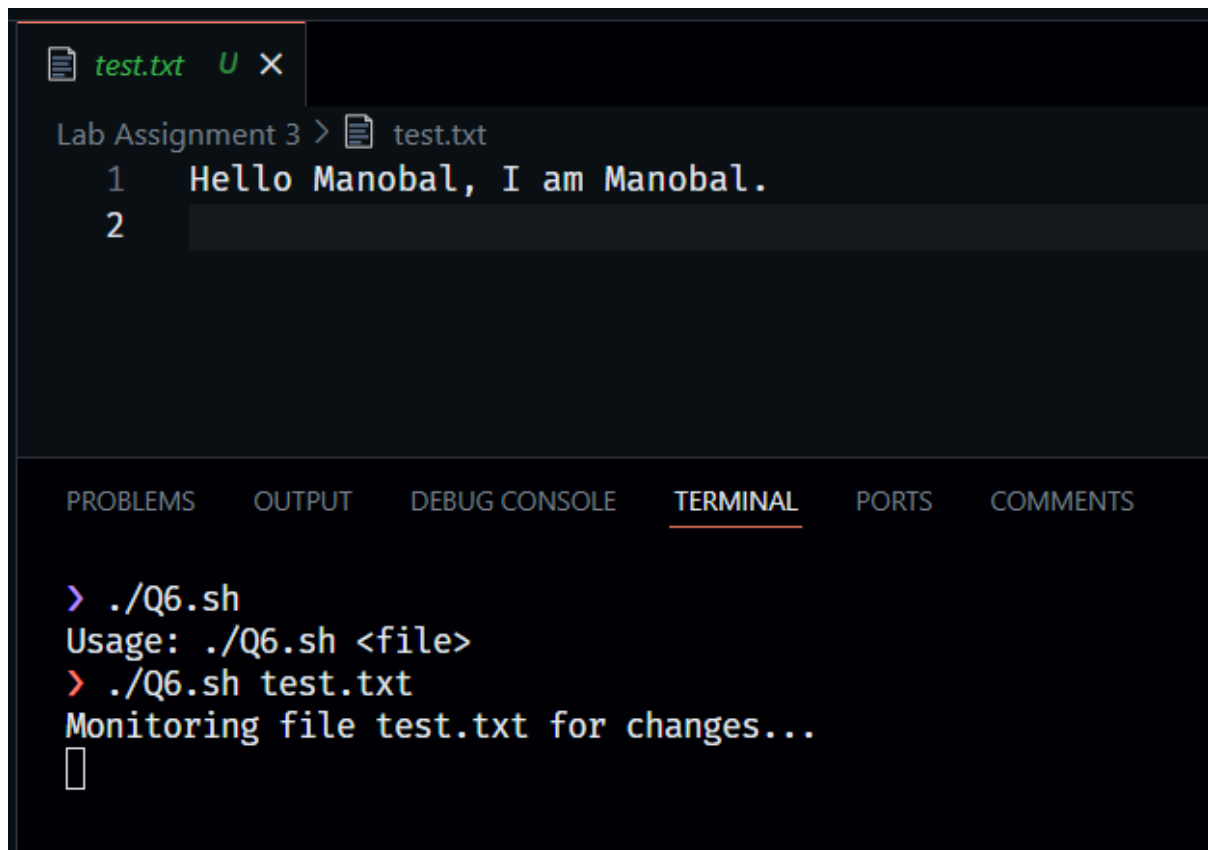
echo "Monitoring file $file for changes..."

# 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
```

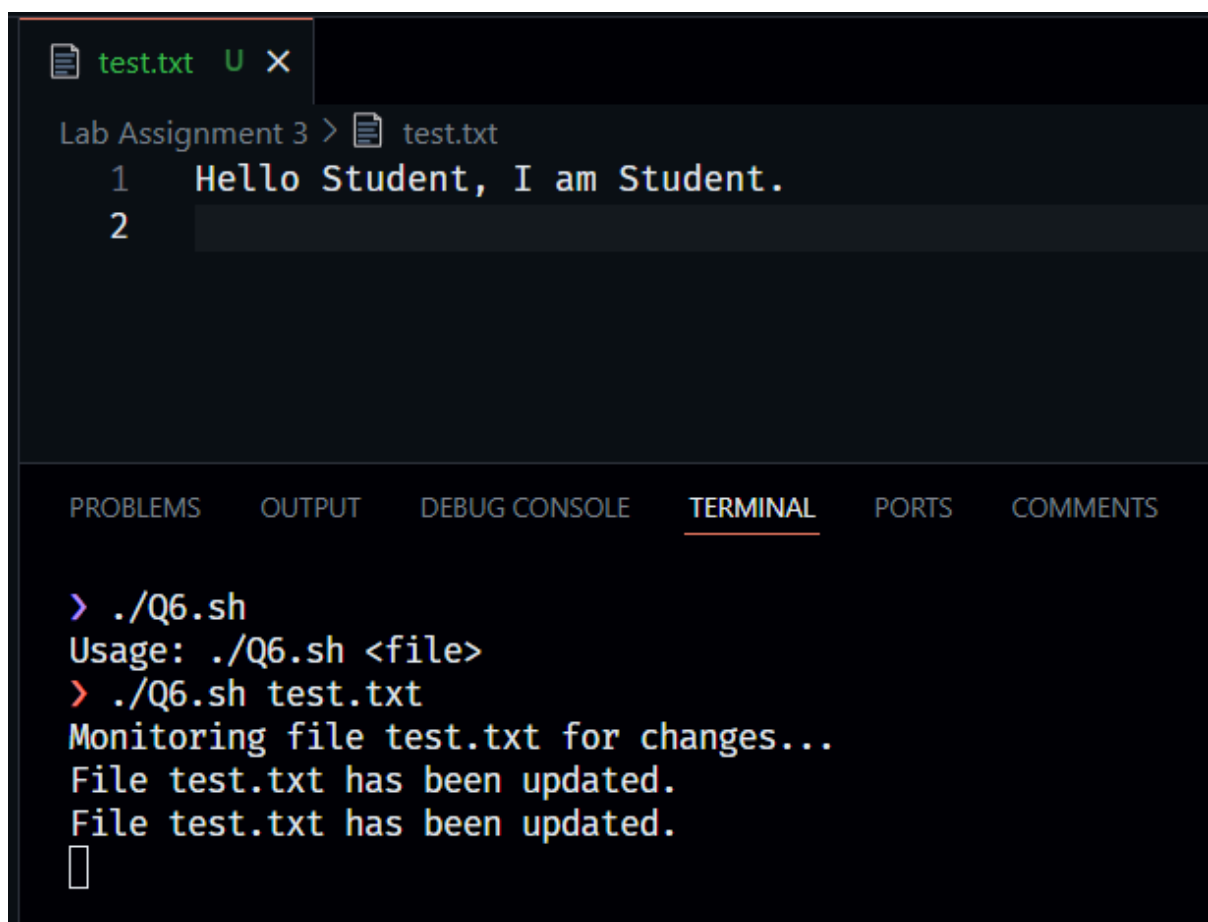
Output:



```
test.txt U X
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 X
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:

```
#!/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

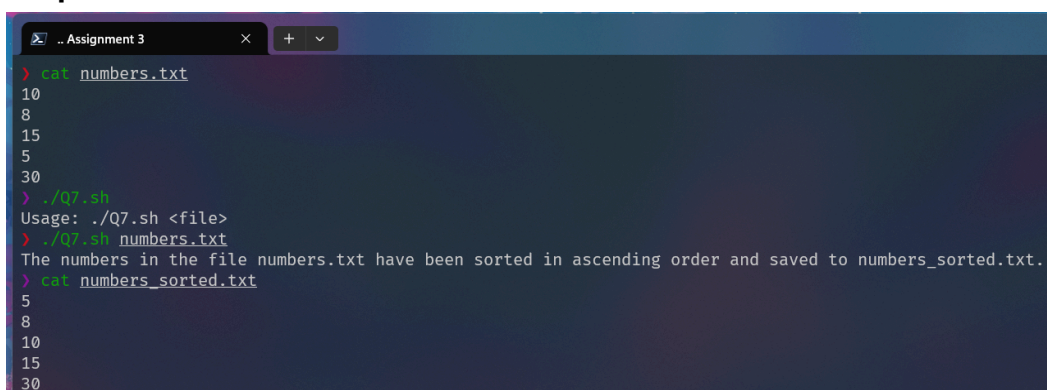
# Check if the file exists
if [ ! -f $file ]; then
    echo "The file $file does not exist."
    exit 1
fi

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:



A terminal window titled "Assignment 3" showing the execution of a shell script. The user first displays the contents of "numbers.txt", which are the numbers 10, 8, 15, 5, 30. Then, they run the script with the file name. The script outputs a usage message and then a confirmation message stating that the numbers have been sorted and saved to "numbers_sorted.txt". Finally, the user displays the contents of the new file, which are the numbers 5, 8, 10, 15, 30, sorted in ascending order.

```
.. Assignment 3 x + v
> cat numbers.txt
10
8
15
5
30
> ./Q7.sh
Usage: ./Q7.sh <file>
> ./Q7.sh numbers.txt
The numbers in the file numbers.txt have been sorted in ascending order and saved to numbers_sorted.txt.
> cat numbers_sorted.txt
5
8
10
15
30
```

Q8.)

Code:

```
#!/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

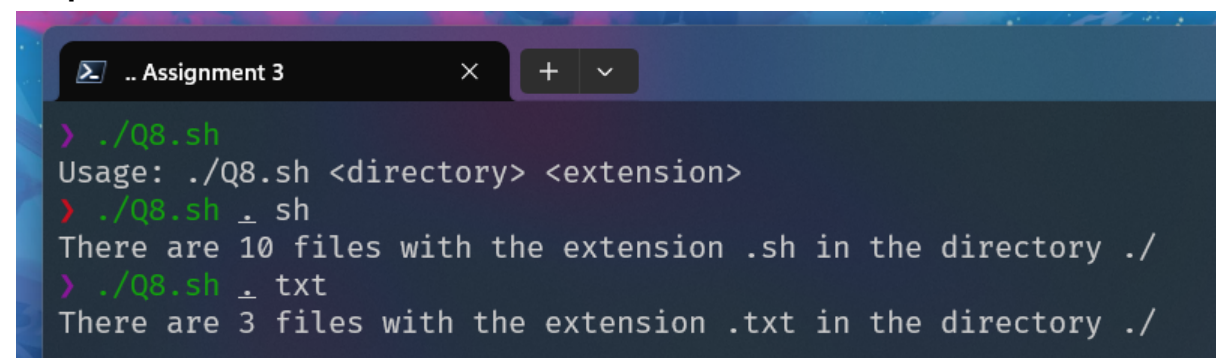
# Get the directory and extension
directory=$1
extension=$2

# 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:



The screenshot shows a terminal window titled "Assignment 3". The user runs the script `./Q8.sh` without arguments, which displays the usage message: `Usage: ./Q8.sh <directory> <extension>`. Then, the user runs `./Q8.sh . sh`, and the script outputs: `There are 10 files with the extension .sh in the directory ./`. Finally, the user runs `./Q8.sh . txt`, and the script outputs: `There are 3 files with the extension .txt in the directory ./`.

Q9.)

Code:

```
#!/bin/bash

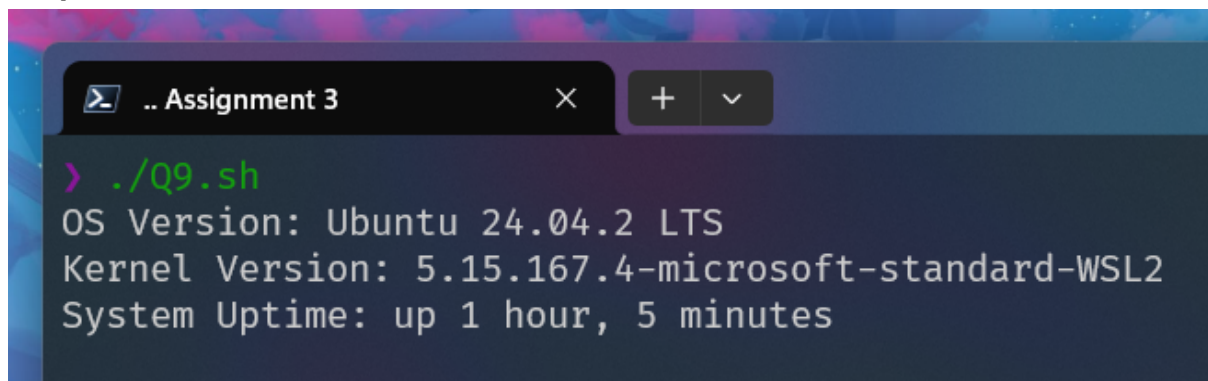
# Question 9
# Script to display system information such as OS version, kernel
version, and system uptime.

# 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:

A screenshot of a terminal window titled '.. Assignment 3'. The terminal shows the execution of the script './Q9.sh' and its output. The output displays the OS version as 'Ubuntu 24.04.2 LTS', the kernel version as '5.15.167.4-microsoft-standard-WSL2', and the system uptime as 'up 1 hour, 5 minutes'.

```
> ./Q9.sh
OS Version: Ubuntu 24.04.2 LTS
Kernel Version: 5.15.167.4-microsoft-standard-WSL2
System Uptime: up 1 hour, 5 minutes
```

Q10.)

Code:

```
#!/bin/bash

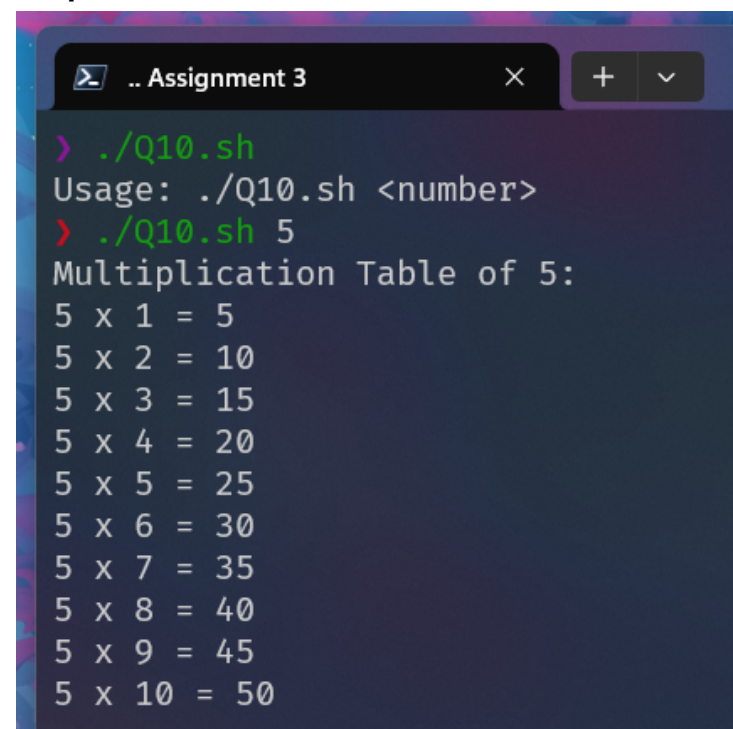
# Question 10
# Script to display multiplication table of a number.

# 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:



```
> ./Q10.sh
Usage: ./Q10.sh <number>
> ./Q10.sh 5
Multiplication Table of 5:
5 x 1 = 5
5 x 2 = 10
5 x 3 = 15
5 x 4 = 20
5 x 5 = 25
5 x 6 = 30
5 x 7 = 35
5 x 8 = 40
5 x 9 = 45
5 x 10 = 50
```