

Exp 4: a)

Write a shell script that determines the period for which a specified user is working on the system.

```
#!/bin/bash
# Script to determine the period for which a specified user is working on the system.

# Check if a user name is provided
if [ $# -eq 0 ]; then
    echo "Usage: $0 <username>"
    exit 1
fi

# Get the user name
user=$1

# Check if the user exists
if ! id "$user" > /dev/null; then
    echo "User '$user' does not exist."
    exit 1
fi

# Get the user's home directory
home=$(getent passwd "$user" | cut -d: -f6)

# Get the user's login shell
shell=$(getent passwd "$user" | cut -d: -f7)

# Get the user's last login date and time
last_login=$(last -t 1 -F | grep "$user" | tail -n 1 | cut -d: -f1,4,6)

# Get the current date and time
current_date=$(date +%Y-%m-%d %H:%M:%S)

# Calculate the difference between the current date and the last login date
diff=$(date -d "$current_date" +%s - $(date -d "$last_login" +%s))

# Convert the difference to hours and minutes
hours=$((diff / 3600))
minutes=$((diff % 3600 / 60))

# Display the result
echo "User '$user' has been working on the system for $hours hours and $minutes minutes."
exit 0
```

```
ayush@ayush-VirtualBox: ~$ ./hash.sh
enter the user name : hash
invalid user
```

B) Write a shell script that displays all the lines between start and end line numberS passed as argument.

```
#!/bin/bash
# Script to display all the lines between start and end line numberS passed as argument.

# Check if two arguments are provided
if [ $# -ne 2 ]; then
    echo "Usage: $0 <start_line> <end_line>"
    exit 1
fi

# Get the start and end line numbers
start=$1
end=$2

# Check if the start line is less than the end line
if [ $start -ge $end ]; then
    echo "Start line must be less than end line."
    exit 1
fi

# Display the lines between start and end line numbers
sed -n "${start},${end}p" /etc/passwd
```

```
ayush@ayush-VirtualBox: ~$ ./hazy.sh
enter the filename : /etc/passwd
enter the starting line number : 1
enter the ending line number : 5
yes
ayush@ayush-VirtualBox: ~$
```

C) Write a shell script that deletes all lines containing a specified word in one or more files supplied as

```
#!/bin/bash
# Script to delete all lines containing a specified word in one or more files supplied as arguments to it.

# Check if at least one argument is provided
if [ $# -lt 1 ]; then
    echo "Usage: $0 <word> <file1> <file2> ..."
    exit 1
fi

# Get the word to be searched
word=$1

# Get the files to be searched
shift

# Loop through the files and delete the lines containing the word
for file in "$@"; do
    sed -i "/$word/d" "$file"
done
```

arguments to it.

```
ayush@ayush-VirtualBox: ~$ ./hazy.sh
enter the filename : /etc/passwd
enter the starting line number : 1
enter the ending line number : 5
yes
ayush@ayush-VirtualBox: ~$
```