LAB 10

Q1.

We can use -f /usr/bin to test if it is a file and -d/usr/bin to check if it is a directory.

Q2.

```
#!/bin/bash

# Prompt the user to enter the first string

echo "Enter the first string:"

read string1

# Prompt the user to enter the second string

echo "Enter the second string:"

read string2

# Compare the two strings

if [ "$string1" = "$string2" ]; then

   echo "The strings are equal."

elif [ "$string1" \< "$string2" ]; then

   echo "String 1 is less than String 2."

else

   echo "String 1 is greater than String 2."

fi
```

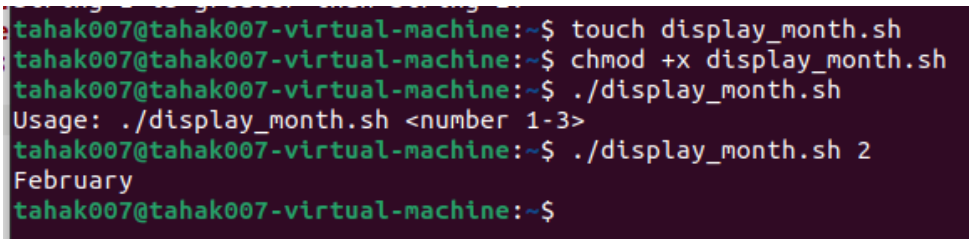
chmod +x compare_strings.sh // making it executable

```
tahak007@tahak007-virtual-machine:~$ touch compare_strings.sh
tahak007@tahak007-virtual-machine:~$ chmod +x compare_strings.sh
tahak007@tahak007-virtual-machine:~$ ./compare_strings.sh
Enter the first string:
Milan
Enter the second string:
Arsenal
String 1 is greater than String 2.
tahak007@tahak007-virtual-machine:~$
```

Q3.

```
#!/bin/bash

# Check if a parameter is provided

if [ "$#" -eq 0 ]; then

   echo "Usage: $0 <number 1-3>"

   exit 1

fi
```

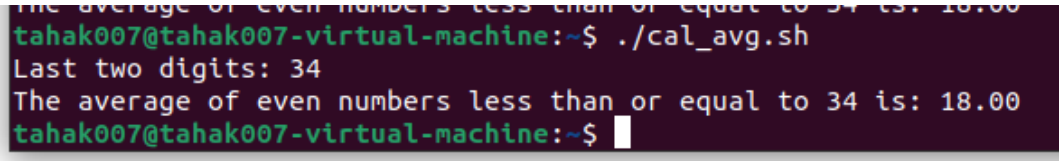
```
# Extract the first parameter
number="$1"
# Use a case statement to display the corresponding month
case "$number" in
  1)
    echo "January"
    ;;
  2)
    echo "February"
    ;;
  3)
    echo "March"
    ;;
  *)
    echo "Invalid input. Please enter a number from 1 to 3."
    exit 1
    ;;
esac
```



Q4.

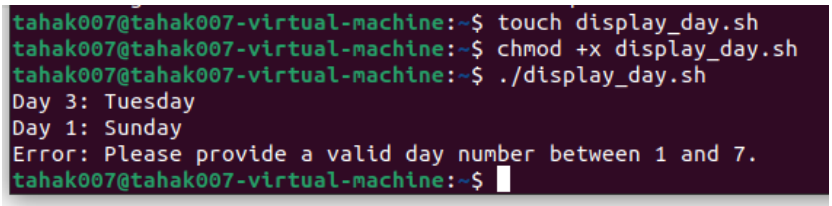
```
#!/bin/bash
# Extract the last two digits from the roll number
last_two_digits=$(echo "CS-22134" | grep -oE "[0-9]{2}$")
# Initialize variables
sum=0
count=0
# Calculate the average of even numbers less than or equal to the last two digits
for ((i = 2; i <= last_two_digits; i += 2)); do
  sum=$((sum + i))
```

```
    count=$((count + 1))
done
# Check if there are even numbers to calculate the average
if [ "$count" -gt 0 ]; then
    average=$((sum / count))
    echo "The average of even numbers less than or equal to $last_two_digits is: $average"
else
    echo "There are no even numbers less than or equal to $last_two_digits."
fi
```



Q5.

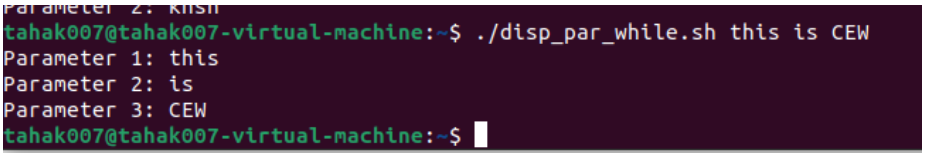
```
#!/bin/bash
# Function to display the day of the week
display_day() {
    # Map day numbers to day names
    days=("Sunday" "Monday" "Tuesday" "Wednesday" "Thursday" "Friday" "Saturday")
    # Check if the provided day number is valid (between 1 and 7)
    if [ "$1" -ge 1 ] && [ "$1" -le 7 ]; then
        echo "Day $1: ${days[$1 - 1]}"
    else
        echo "Error: Please provide a valid day number between 1 and 7."
    Fi }
# Example usage:
# Call the function with a day number as an argument
display_day 3
display_day 1
display_day 8
```



Q6.

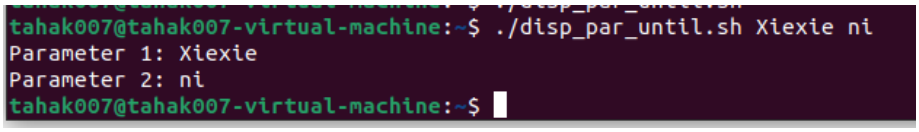
Using While

```
#!/bin/bash

# Using while statement to display parameters and their numbers

count=1

while [ "$#" -gt 0 ]; do

   echo "Parameter $count: $1"

   shift

   ((count++))

done
```



Using until

```
#!/bin/bash

# Using until statement to display parameters and their numbers

count=1

until [ "$#" -eq 0 ]; do

   echo "Parameter $count: $1"

   shift

   ((count++))

done
```



Q7.

```
#!/bin/bash

while true; do

   # Display menu

   echo "Menu:"

   echo "1. Quotient"

   echo "2. Remainder"
```

```
echo "3. Quit"

# Prompt user for choice
read -p "Enter your choice (1-3): " choice
case "$choice" in
  1)
    # Quotient
    read -p "Enter the dividend: " dividend
    read -p "Enter the divisor: " divisor
    if [ "$divisor" -eq 0 ]; then
      echo "Error: Division by zero is not allowed."
    else
      result=$((dividend / divisor))
      echo "Quotient: $result"
      break
    fi
    ;;
  2)
    # Remainder
    read -p "Enter the dividend: " dividend
    read -p "Enter the divisor: " divisor
    if [ "$divisor" -eq 0 ]; then
      echo "Error: Division by zero is not allowed."
    else
      result=$((dividend % divisor))
      echo "Remainder: $result"
      break
    fi
    ;;
  3)
    # Quit
    echo "Exiting the script."
```

```
        exit 0
        ;;
    *)
        # Invalid choice
        echo "Invalid choice. Please choose a number from 1 to 3."
        ;;
  esac
done
```

```
Menu:
1. Quotient
2. Remainder
3. Quit
Enter your choice (1-3): 2
Enter the dividend: 145
Enter the divisor: 3
Remainder: 1
tahak007@tahak007-virtual-machine:~$ ./div_menu.sh
Menu:
1. Quotient
2. Remainder
3. Quit
Enter your choice (1-3): 1
Enter the dividend: 156
Enter the divisor: 14
Quotient: 11
tahak007@tahak007-virtual-machine:~$
```