LAB 10

1. **What test command should be used to test that /usr/bin is a directory or a file?**

For file, use -f test command [-d “/usr/bin”]
For directory, use -d test command [-f “/usr/bin”]

2. **Write a script that takes two strings as input compares them and depending upon the results of the comparison prints the results.**

```
#!/bin/bash

echo "Enter the first string: " read str1

echo "Enter the second string: " read str2

if [ "$str1" = "$str2" ]; then

        echo "The strings are equal"

else

        echo "The strings are not equal"

fi
```

3. **Write a script that takes a number (parameter) from 1-3 as input and uses case to display the name of corresponding month.**

```
#!/bin/bash

case "$1" in

1)      echo "January";;

2)      echo "February";;

3)      echo "March";;

*) echo "Only enter numbers from 1-3";;
esac
```

4. **Write a script that calculates the average of all even numbers less than or equal to your roll number and prints the result.**

```
#!/bin/bash

roll=1
sum=0
count=0

for ((i=2; i<=roll; i+=2)); do
        sum=$((sum + i))
        count=$((count + 1))
done

if [ "$count" -gt 0 ]; then

        average=$((sum/count))
        echo "The Average of even numbers upto $roll: $average"

else

        echo "There is no roll even number uptill your roll number :("

fi
```

5. **Write a function that displays the name of the week days starting from Sunday if the user passes a day number. If a number provided is not between 1 and 7 an error message is displayed.**

```
#!/bin/bash
func (){
        case "$1" in

        1)      echo "Sunday";;

        2)      echo "Monday";;

        3)      echo "Tuesday";;

        4)      echo "Wednesday";;

        5)      echo "Thursday";;

        6)      echo "Friday";;

        7)      echo "Saturday";;


        *) echo "Only enter numbers from 1-7";;
        esac

 } func $1
```

6. **Write scripts that displays the parameters passed along with the parameter number using while and until statements.**

USING WHILE

```
#!/bin/bash

count=1

while [ "$1" ]
do
        echo "Argument # $count = $1 "
        count=$((count+1))
        shift
done
```

USING UNTIL

```
 #!/bin/bash

count=1

until [ ! "$1" ]
do
        echo "Argument # $count = $1 "
        count=$((count+1))
        shift
done
```

7. Write a script that displays the following menu:

- **Quotient**
- **Remainder**

Depending on user's choice, the result of division must be displayed and the loop breaks. The two numbers (dividend and divisor) must be supplied at runtime as command line arguments. If user chooses an item that is not in the list, he must be prompted to make proper choice and the loop must restart (or continue).

```
#!/bin/bash

while true; do

        echo "Menu:"
        echo "1. Quotient"
        echo "2. Remainder"
        echo "3. Exit"
        read -p "Enter your choice (1/2/3): " choice

case $choice in
        1)
          read -p "Enter dividend: " dividend
          read -p "Enter divisor: " divisor
          if [ "$divisor" -eq 0 ]; then
            echo "Error: Division by zero is not allowed."
          else
            quotient=$((dividend / divisor))
            echo "Quotient: $quotient"        break
          fi;;

        2)
          read -p "Enter dividend: " dividend
          read -p "Enter divisor: " divisor
          if [ "$divisor" -eq 0 ]; then
            echo "Error: Division by zero is not allowed."
        else
            remainder=$((dividend % divisor))
            echo "Remainder: $remainder"
            break
          fi;;

        3)
          echo "Exiting the script."
          exit 0;;
        *)
          echo "Invalid choice. Please choose 1, 2, or 3.";;
  esac
done
```