MODULE 4

ASSIGNMENT 1

Logical Operators

1) Check whether the file exists and is executable using logical operators.

Hint:man test

Open file.sh ~/ Save

file.sh wordcount.sh

```
#!/bin/bash

read -p "Enter path of the file: " file_path

if [ -x "$file_path" ] && [ -e "$file_path" ]; then
    echo "File exists and is executable."
else
    echo "File does not exist or is not executable."
fi

```

```
vboxuser@Ubuntu:~$ gedit file.sh
vboxuser@Ubuntu:~$ chmod +x file.sh
vboxuser@Ubuntu:~$ bash file.sh
File exists and is executable.
vboxuser@Ubuntu:~$ bash file.sh
Enter path of the file: /home/vboxuser/exercise.txt
File exists and is executable.
vboxuser@Ubuntu:~$ bash file.sh
Enter path of the file: /home/vboxuser/civil.txt
File exists and is executable.
vboxuser@Ubuntu:~$ ls -l civi.txt
ls: cannot access 'civi.txt': No such file or directory
vboxuser@Ubuntu:~$ ls -l civil.txt
-rwxrwxr-x 1 vboxuser vboxuser 135 Sep 25 10:32 civil.txt
vboxuser@Ubuntu:~$ bash file.sh
Enter path of the file: /home/vboxuser/test3
File does not exist or is not executable.
vboxuser@Ubuntu:~$
```

ASSIGNMENT 2

Arithmetic Comparison

1) Write a program to demonstrate the use of not equal to operator.

Hint: -ne

file.sh | wordcount.sh

```
#!/bin/bash

read -p "Enter filename: " y

x=`cat $y | wc -w`

if [ $x -eq 0 ]; then
   echo "the file $y has zero words"
elif [ $x -ne 0 ]; then
   echo "the file $y has $x words"
else
   echo "invalid"
fi
```

```
vboxuser@Ubuntu:~$ gedit wordcount.sh
vboxuser@Ubuntu:~$ chmod +x wordcount.sh
vboxuser@Ubuntu:~$ bash wordcount.sh
Enter filename: array2.sh
the file array2.sh has 59 words
vboxuser@Ubuntu:~$ bash wordcount.sh
Enter filename: file.sh
the file file.sh has 38 words
vboxuser@Ubuntu:~$ bash wordcount.sh
Enter filename: names.txt
the file names.txt has 24 words
vboxuser@Ubuntu:~$ bash wordcount.sh
Enter filename: seddemo.txt
the file seddemo.txt has 25 words
vboxuser@Ubuntu:~$
```

ASSIGNMENT 3

String and File attributes

1) Explore some more attributes

-r

-x

-o

```
vboxuser@Ubuntu: ~
vboxuser@Ubuntu:~$ gedit strcompare.sh
vboxuser@Ubuntu:~$ chmod +x strcompare.sh
vboxuser@Ubuntu:~$ whoami
vboxuser
vboxuser@Ubuntu:~$ bash strcompare.sh
You have no permission to run strcompare.sh as non-root user.
vboxuser@Ubuntu:~$ sudo ./strcompare.sh
[sudo] password for vboxuser:
welcome root!.
vboxuser@Ubuntu:~$ ls -o strcompare.sh
-rwxrwxr-x 1 vboxuser 149 Sep 28 16:50 strcompare.sh
vboxuser@Ubuntu:~$ ls -r strcompare.sh
strcompare.sh
vboxuser@Ubuntu:~$ ls -x strcompare.sh
strcompare.sh
vboxuser@Ubuntu:~$ touch unix.txt
vboxuser@Ubuntu:~$ ls -o strcompare.sh
-rwxrwxr-x 1 vboxuser 149 Sep 28 16:50 strcompare.sh
vboxuser@Ubuntu:~$ ls -o unix.txt
-rw-rw-r-- 1 vboxuser 0 Sep 28 17:00 unix.txt
vboxuser@Ubuntu:~$ ls -r unix.txt
unix.txt
vboxuser@Ubuntu:~$ ls -w unix.txt
ls: invalid line width: 'unix.txt'
vboxuser@Ubuntu:~$ ls -x unix.txt
unix.txt
vboxuser@Ubuntu:~$ ls -s unix.txt
0 unix.txt
vboxuser@Ubuntu:~$
```

ASSIGNMENT 4

Conditional Loops

1) **Find the sum of first n prime numbers.**

prime.sh
~/

```bash
#!/bin/bash

read -p "Enter the limit to print prime numbers: " n

sum=0
count=0
number=2

while [ $count -lt $n ]; do
    is_prime=true

    for ((i = 2; i * i <= number; i++)); do
        if [ $((number % i)) -eq 0 ]; then
            is_prime=false
            break
        fi
    done

    if $is_prime; then
        sum=$((sum + number))
        count=$((count + 1))
    fi

    number=$((number + 1))
done

echo "The sum of the first $n prime numbers is: $sum"
```

```
vboxuser@Ubuntu:~$ bash prime.sh
Enter the limit to print prime numbers: 4
The sum of the first 4 prime numbers is: 17
vboxuser@Ubuntu:~$ bash prime.sh
Enter the limit to print prime numbers: 5
The sum of the first 5 prime numbers is: 28
```

ASSIGNMENT 5

More on Loops

1) Retype nested-for.sh bash script using nested while loop

2) Save your program with the name: nested-while.sh

Open nestedwhile.sh ~/ Save

prime.sh | nestedwhile.sh

```
for dir in test*; do
    echo "Files in $dir directory:"
    echo ""
    for file in $(ls -1 $dir); do
        echo $file
    done
    echo "------------------------"
done
```

```
vboxuser@Ubuntu:~/simple-nested-while$ ./nestedwhile.sh
Files in test directory:

file
file1
file2
file3
------------------------
Files in test1 directory:

file
file1
file2
file3
------------------------
Files in test2 directory:

file
file1
file2
file3
------------------------
Files in test3 directory:

file
file1
file2
file3
------------------------
```

ASSIGNMENT 6

1) Write a menu driven program for mathematical calculation

a. It should take user inputs a and b

b. It should ask for mathematical operator (+, -, / and *).

c. Do the calculation

d. Print the output

Open calculator.sh ~/ Save

prime.sh | calculator.sh

```bash
#!/bin/bash

while true; do
    echo "Mathematical Calculator Menu:"
    echo "1. Addition (+)"
    echo "2. Subtraction (-)"
    echo "3. Multiplication (*)"
    echo "4. Division (/)"
    echo "5. Exit"
    read -p "Choose an operation (1/2/3/4/5): " choice

    case $choice in
        1)
            read -p "Enter the first number (a): " a
            read -p "Enter the second number (b): " b
            result=$((a + b))
            echo "Result: $a + $b = $result"
            ;;
        2)
            read -p "Enter the first number (a): " a
            read -p "Enter the second number (b): " b
            result=$((a - b))
            echo "Result: $a - $b = $result"
            ;;
        3)
            read -p "Enter the first number (a): " a
            read -p "Enter the second number (b): " b
            result=$((a * b))
            echo "Result: $a * $b = $result"
            ;;

        4)
            read -p "Enter the first number (a): " a
            read -p "Enter the second number (b): " b
            if [ "$b" -eq 0 ]; then
                echo "Error: Division by zero is not allowed."
            else
                result=$(echo "scale=2; $a / $b" | bc)
                echo "Result: $a / $b = $result"
            fi
            ;;
        5)
            echo "Exiting the calculator."
            exit 0
            ;;
        *)
            echo "Invalid choice. Please select a valid option (1/2/3/4/5)."
            ;;
    esac
done
```

```
vboxuser@Ubuntu:~$ gedit calculator.sh
vboxuser@Ubuntu:~$ chmod +x calculator.sh
vboxuser@Ubuntu:~$ ./calculator.sh
Mathematical Calculator Menu:
1. Addition (+)
2. Subtraction (-)
3. Multiplication (*)
4. Division (/)
5. Exit
Choose an operation (1/2/3/4/5): 1
Enter the first number (a): 2
Enter the second number (b): 5
Result: 2 + 5 = 7
Mathematical Calculator Menu:
1. Addition (+)
2. Subtraction (-)
3. Multiplication (*)
4. Division (/)
5. Exit
Choose an operation (1/2/3/4/5): 2
Enter the first number (a): 45
Enter the second number (b): 24
Result: 45 - 24 = 21
Mathematical Calculator Menu:
1. Addition (+)
2. Subtraction (-)
3. Multiplication (*)
4. Division (/)
5. Exit
Choose an operation (1/2/3/4/5): 3
Enter the first number (a): 4
Enter the second number (b): 5
Result: 4 * 5 = 20
```

```
Mathematical Calculator Menu:
1. Addition (+)
2. Subtraction (-)
3. Multiplication (*)
4. Division (/)
5. Exit
Choose an operation (1/2/3/4/5): 5
Exiting the calculator.
```

ASSIGNMENT 7

Using File Descriptors

1) Try to append few lines to a file test.txt using file descriptor.

2) Display the content of the file using file descriptor.

Open | append.sh ~/ | Save

append.sh | test.txt | display.sh

```bash
#!/bin/bash
exec 3>>test.txt

echo "This is line 1" >&3
echo "This is line 2" >&3
echo "This is line 3" >&3


exec 3>&-

```

Open | test.txt ~/ | Save

append.sh | test.txt | display.sh

```
This is line 1
This is line 2
This is line 3
```

append.sh × test.txt × display.sh ×

```
#!/bin/bash
exec 4<test.txt


while IFS= read -r line; do
    echo "$line"
done <&4


exec 4<&-

```

```
vboxuser@Ubuntu:~$ gedit append.sh
vboxuser@Ubuntu:~$ chmod +x append.sh
vboxuser@Ubuntu:~$ ./append.sh
vboxuser@Ubuntu:~$ gedit test.txt
vboxuser@Ubuntu:~$ gedit display.sh
vboxuser@Ubuntu:~$ chmod +x display.sh
vboxuser@Ubuntu:~$ ./display.sh
This is line 1
This is line 2
This is line 3
vboxuser@Ubuntu:~$
```

ASSIGNMENT 8

Basics of functions

1) Write a program with two functions:

a. The first function should display diskspace usage in human readable form.

(Hint: df -h)

b. The second function should display filesystem usage in human readable form.

(Hint: du -h)

Open func.sh ~/ Save

append.sh func.sh

```bash
#!/bin/bash

function displayDiskSpaceUsage() {
    echo "Disk Space Usage:"
    df -h
}

function displayFilesystemUsage() {
    echo "Filesystem Usage:"
    du -h
}

displayDiskSpaceUsage
echo "------------------------"
displayFilesystemUsage
```

```
vboxuser@Ubuntu:~$ gedit func.sh
vboxuser@Ubuntu:~$ chmod +x func.sh
vboxuser@Ubuntu:~$ ./func.sh
Disk Space Usage:
Filesystem      Size  Used Avail Use% Mounted on
tmpfs           352M  1.6M  351M   1% /run
/dev/sda3        24G   14G  9.5G  59% /
tmpfs           1.8G     0  1.8G   0% /dev/shm
tmpfs           5.0M  4.0K  5.0M   1% /run/lock
/dev/sda2       512M  6.1M  506M   2% /boot/efi
tmpfs           352M  128K  352M   1% /run/user/1000
/dev/sr0         52M   52M     0 100% /media/vboxuser/VBox_GAs_7.0.10
------------------------
Filesystem Usage:
4.0K    ./Public
8.0K    ./.cache/mesa_shader_cache/69
12K     ./.cache/mesa_shader_cache/a3
8.0K    ./.cache/mesa_shader_cache/c9
8.0K    ./.cache/mesa_shader_cache/5e
8.0K    ./.cache/mesa_shader_cache/1a
12K     ./.cache/mesa_shader_cache/80
8.0K    ./.cache/mesa_shader_cache/ef
16K     ./.cache/mesa_shader_cache/45
```

ASSIGNMENT 9

More on functions

1) Write a program,

a. where the function accepts two arguments.

b. The function should multiply the two arguments.

c. Make 3 function calls with arguments - (1, 2), (2, 3) and (3, 4)

Open func1.sh ~/ Save

append.sh func1.sh

```
#!/bin/bash

function multiply() {
    local result=$(( $1 * $2 ))
    echo "Multiplying $1 by $2 equals $result"
}

multiply 1 2
multiply 2 3
multiply 3 4

```

```
vboxuser@Ubuntu:~$ gedit func1.sh
vboxuser@Ubuntu:~$ chmod func1.sh
chmod: missing operand after ‘func1.sh’
Try 'chmod --help' for more information.
vboxuser@Ubuntu:~$ chmod +x func1.sh
vboxuser@Ubuntu:~$  ./func1.sh
Multiplying 1 by 2 equals 2
Multiplying 2 by 3 equals 6
Multiplying 3 by 4 equals 12
vboxuser@Ubuntu:~$
```

ASSIGNMENT 10

Arrays and functions

1) Write a program,

a. Where a function adds all the elements in an array.

b. The function should display the sum of elements.

c. Make 2 function calls with array elements- (1, 2, 3) and (4, 5, 6).

Open func2.sh ~/ Save

append.sh | func1.sh | func2.sh

```bash
#!/bin/bash

function addArrayElements() {
    local sum=0
    for element in "${@}"; do
        sum=$((sum + element))
    done
    echo "Sum of elements: $sum"
}

addArrayElements 1 2 3

addArrayElements 4 5 6
```

```
vboxuser@Ubuntu:~$ gedit func2.sh
vboxuser@Ubuntu:~$ chmod +x func2.sh
vboxuser@Ubuntu:~$ ./func2.sh
Sum of elements: 6
Sum of elements: 15
vboxuser@Ubuntu:~$
```