Nithin S
221IT085

IT253 Lab Assignment 2

Shell Programming

1. Write Script to see current date, time, username, and current directory

```
# !/bin/bash

echo "Current date: $(date '+%d-%m-%Y')"
echo "Current time: $(date '+%H:%M:%S')"
echo "Username: $USER"
echo "Current directory: $(pwd)"
```

OUTPUT

```
student@HP-Elite600G9-08:~/Desktop/assgn$ chmod +x script1
student@HP-Elite600G9-08:~/Desktop/assgn$ ./script1
Current date: 15-01-2024
Current time: 15:30:21
Username: student
Current directory: /home/student/Desktop/assgn
student@HP-Elite600G9-08:~/Desktop/assgn$
```

2. How to write shell script that will add two numbers, which are supplied as command line argument, and if this two numbers are not given show error and its usage

```
# !/bin/bash

if [ $# -eq 2 ]
then
        sum=$(($1 + $2))
        echo "Sum of $1 and $2 is: $sum"
else
        echo "ERROR:You have not entered two numbers"
fi
```

OUTPUT

```
student@HP-Elite600G9-08:~/Desktop/assgn$ chmod +x script2
student@HP-Elite600G9-08:~/Desktop/assgn$ ./script2
ERROR:You have not entered two numbers
student@HP-Elite600G9-08:~/Desktop/assgn$ ./script2 3
ERROR:You have not entered two numbers
student@HP-Elite600G9-08:~/Desktop/assgn$ ./script2 3 4
Sum of 3 and 4 is: 7
student@HP-Elite600G9-08:~/Desktop/assgn$
```

Uses of Command line arguments:

Customize script behavior by passing values during execution.
Control script flow with different options and switches.
Pass filenames or input data as arguments for processing.
Execute script parts based on provided options or flags.
Process each argument in a loop using "$@".
Display help or usage information with specific options.
Override default parameters using command-line arguments.

3.Write Script to find out biggest number from given three nos. Numbers are supplied as commandline argument. Print error if sufficient arguments are not supplied

```
# !/bin/bash

if [ $# -eq 3 ]
then
        if [ $1 -ge $2 ] && [ $1 -ge $3 ]
        then
                echo "The greatest of three number entered is $1"

        elif [ $2 -ge $1 ] && [ $2 -ge $3 ]
        then
                echo "The greatest of three number entered is $2"

        elif [ $3 -gt $1 ] && [ $3 -gt $1 ]
        then
                echo "The greatest of three number entered is $3"
fi
else
        echo "ERROR:You have not entered three numbers"
fi
```

OUTPUT

```
student@HP-Elite600G9-08:~/Desktop/assgn$ chmod +x script3
student@HP-Elite600G9-08:~/Desktop/assgn$ ./script3
ERROR:You have not entered three numbers
student@HP-Elite600G9-08:~/Desktop/assgn$ ./script3 1 2
ERROR:You have not entered three numbers
student@HP-Elite600G9-08:~/Desktop/assgn$ ./script3 1 2 3
The greatest of three number entered is 3
student@HP-Elite600G9-08:~/Desktop/assgn$ ./script3 500 5 5
The greatest of three number entered is 500
student@HP-Elite600G9-08:~/Desktop/assgn$ ./script3 99 99 97
The greatest of three number entered is 99
student@HP-Elite600G9-08:~/Desktop/assgn$ ./script3 54 67 8
The greatest of three number entered is 67
student@HP-Elite600G9-08:~/Desktop/assgn$ ./script3 99 99 97 77
ERROR:You have not entered three numbers
student@HP-Elite600G9-08:~/Desktop/assgn$ ./script3 0 0 0
The greatest of three number entered is 0
student@HP-Elite600G9-08:~/Desktop/assgn$ ./script3 1 1 0
The greatest of three number entered is 1
student@HP-Elite600G9-08:~/Desktop/assgn$ ./script3 1 1 1
The greatest of three number entered is 1
student@HP-Elite600G9-08:~/Desktop/assgn$ ./script3 1 1 1 0
ERROR:You have not entered three numbers
student@HP-Elite600G9-08:~/Desktop/assgn$
```

4. Write script to print the following numbers as 5,4,3,2,1 using while loop.

```
# !/bin/bash

c=5
while [ $c -ge 1 ]
do
        echo -n "$c "
        (( c-- ))
done
```

OUTPUT

```
student@HP-Elite600G9-08:~/Desktop/assgn$ chmod +x script4
student@HP-Elite600G9-08:~/Desktop/assgn$ ./script4
5 4 3 2 1 student@HP-Elite600G9-08:~/Desktop/assgn$
```

5. Write Script, using case statement to perform basic math operation as follows: + addition, - subtraction, x multiplication, / division

```
# !/bin/bash

echo "Enter Two numbers : "
read a
read b

echo "Enter Choice :"
echo "1. Addition"
echo "2. Subtraction"
echo "3. Multiplication"
echo "4. Division"
read ch


case $ch in
1)res=`echo $a + $b | bc`
;;
2)res=`echo $a - $b | bc`
;;
3)res=`echo $a \* $b | bc`
;;
4)res=`echo "scale=2; $a / $b" | bc`
;;
esac
echo "Result : $res"
```

OUTPUT

```
nithin@nithin1729s:~/Codes/Sem 4/IT253/Lab/Lab_2$ chmod +x script5
nithin@nithin1729s:~/Codes/Sem 4/IT253/Lab/Lab_2$ ./script5
Enter Two numbers :
4
5
Enter Choice :
1. Addition
2. Subtraction
3. Multiplication
4. Division
1
Result : 9
nithin@nithin1729s:~/Codes/Sem 4/IT253/Lab/Lab_2$ ./script5
Enter Two numbers :
3
0
Enter Choice :
1. Addition
2. Subtraction
3. Multiplication
4. Division
4
Error: Division by zero is not allowed.
nithin@nithin1729s:~/Codes/Sem 4/IT253/Lab/Lab_2$ ./script5
Enter Two numbers :
9
0
Enter Choice :
1. Addition
2. Subtraction
3. Multiplication
4. Division
3
Result : 0
nithin@nithin1729s:~/Codes/Sem 4/IT253/Lab/Lab_2$ ./script5
Enter Two numbers :
0
0
Enter Choice :
1. Addition
2. Subtraction
3. Multiplication
4. Division
4
Error: Division by zero is not allowed.
```

6. Write script to print given number in reverse order, for eg. If no is 123 it must print as 321.
(This code takes input as command line args)

```
#!/bin/bash

if [ "$#" -ne 1 ]; then
    echo "Enter a valid number"
    exit 1
fi

number=$1
reversed=""

while [ "$number" -gt 0 ]; do
    digit=$((number % 10))
    reversed="${reversed}${digit}"
    number=$((number / 10))
done

echo "Original Number: $1"

if [ -z "$reversed" ]; then
    echo "Reversed Number: $1"
else
    echo "Reversed Number: $reversed"
fi
```

OUTPUT

```
nithin@nithin1729s:~/Codes/Sem 4/IT253/Lab/Lab_2$ chmod +x script6
nithin@nithin1729s:~/Codes/Sem 4/IT253/Lab/Lab_2$ ./script6 0
Original Number: 0
Reversed Number: 0
nithin@nithin1729s:~/Codes/Sem 4/IT253/Lab/Lab_2$ ./script6 785678
Original Number: 785678
Reversed Number: 876587
nithin@nithin1729s:~/Codes/Sem 4/IT253/Lab/Lab_2$ ./script6 123
Original Number: 123
Reversed Number: 321
nithin@nithin1729s:~/Codes/Sem 4/IT253/Lab/Lab_2$ ./script6 1
Original Number: 1
Reversed Number: 1
nithin@nithin1729s:~/Codes/Sem 4/IT253/Lab/Lab_2$ ./script6 100
Original Number: 100
Reversed Number: 001
nithin@nithin1729s:~/Codes/Sem 4/IT253/Lab/Lab_2$ ./script6 987
Original Number: 987
Reversed Number: 789
```

7. Write script to print given numbers sum of all digit, For eg. If no is 123 it's sum of all digit will be 1+2+3 = 6.(This code takes input as command line args)

```
#!/bin/bash

if [ "$#" -ne 1 ]; then
    echo "Enter a valid number"
    exit 1
fi

number=$1
sum=0

while [ "$number" -gt 0 ];
do
    digit=$((number % 10))
    sum=$((sum + digit))
    number=$((number / 10))
done

echo "Original Number: $1"
echo "Sum of Digits: $sum"
```

OUTPUT

```
nithin@nithin1729s:~/Codes/Sem 4/IT253/Lab/Lab_2$ chmod +x script7
nithin@nithin1729s:~/Codes/Sem 4/IT253/Lab/Lab_2$ ./script7 123
Original Number: 123
Sum of Digits: 6
nithin@nithin1729s:~/Codes/Sem 4/IT253/Lab/Lab_2$ ./script7 100
Original Number: 100
Sum of Digits: 1
nithin@nithin1729s:~/Codes/Sem 4/IT253/Lab/Lab_2$ ./script7
Enter a valid number
nithin@nithin1729s:~/Codes/Sem 4/IT253/Lab/Lab_2$ ./script7 987
Original Number: 987
Sum of Digits: 24
nithin@nithin1729s:~/Codes/Sem 4/IT253/Lab/Lab_2$ ./script7 909
Original Number: 909
Sum of Digits: 18
nithin@nithin1729s:~/Codes/Sem 4/IT253/Lab/Lab_2$ ./script7 500
Original Number: 500
Sum of Digits: 5
nithin@nithin1729s:~/Codes/Sem 4/IT253/Lab/Lab_2$
```

8. Write script to determine whether given file exist or not, file name is supplied as command line argument, also check for sufficient number of command line argument

```
#!/bin/bash


if [ "$#" -ne 1 ]; then
    echo "Enter a valid filename"
    exit 1
fi

file_path=$1

if [ -e "$file_path" ]; then
    echo "The file '$file_path' exists."
else
    echo "The file '$file_path' does not exist."
fi
```

OUTPUT

```
nithin@nithin1729s:~/Codes/Sem 4/IT253/Lab/Lab_2$ chmod +x script8
nithin@nithin1729s:~/Codes/Sem 4/IT253/Lab/Lab_2$ ./script8
Enter a valid filename
nithin@nithin1729s:~/Codes/Sem 4/IT253/Lab/Lab_2$ ./script8 file.txt
The file 'file.txt' does not exist.
nithin@nithin1729s:~/Codes/Sem 4/IT253/Lab/Lab_2$ ./script8 script4
The file 'script4' exists.
nithin@nithin1729s:~/Codes/Sem 4/IT253/Lab/Lab_2$ ./script8 nithin.txt
The file 'nithin.txt' exists.
nithin@nithin1729s:~/Codes/Sem 4/IT253/Lab/Lab_2$
```

9. Write a shell script takes the name a path (eg: /afs/andrew/course/15/123/handin), and counts all the sub directories (recursively).

```
#!/bin/bash

if [ "$#" -ne 1 ]; then
    echo "Enter valid file path"
    exit 1
fi

count=$(ls -R "$1" | wc -l)
echo "$count"
```

OUTPUT

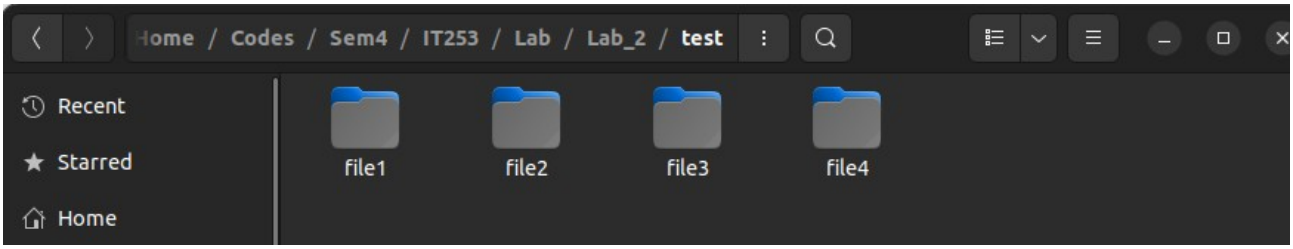
```
nithin@nithin1729s:~/Codes/Sem 4/IT253/Lab/Lab_2$ chmod +x script9
nithin@nithin1729s:~/Codes/Sem 4/IT253/Lab/Lab_2$ ./script9 /home
588723
nithin@nithin1729s:~/Codes/Sem 4/IT253/Lab/Lab_2$ ./script9 /home/nithin/
588720
nithin@nithin1729s:~/Codes/Sem 4/IT253/Lab/Lab_2$
```

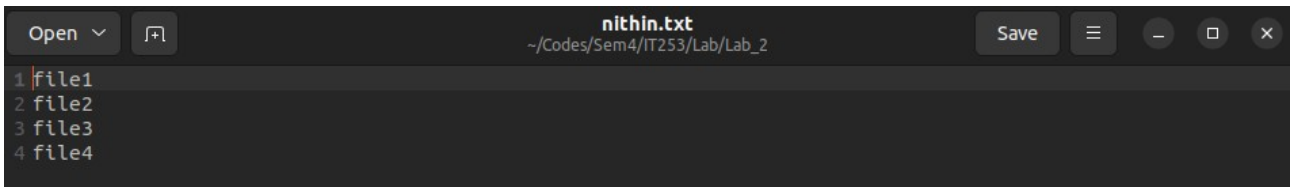
10. Write a shell script that takes a name of a folder as a command line argument, and produce a file that contains the names of all sub folders with size 0 (that is empty sub folders)

```
#!/bin/sh

if [ "$#" -ne 1 ]; then
    echo "Enter valid file path"
    exit 1
fi

ls "$1" |
    while read folder; do
        if [ -d "$1/$folder" ]; then
            files=$(ls "$1/$folder" | wc -l)
            if [ "$files" -eq 0 ]; then
                echo "$folder" >>nithin.txt
            fi
        fi
    done
```

OUTPUT

```
nithin@nithin1729s:~/Codes/Sem4/IT253/Lab/Lab_2$ chmod +x script10
nithin@nithin1729s:~/Codes/Sem4/IT253/Lab/Lab_2$ ./script10
Enter valid file path
nithin@nithin1729s:~/Codes/Sem4/IT253/Lab/Lab_2$ ./script10 test
nithin@nithin1729s:~/Codes/Sem4/IT253/Lab/Lab_2$
```



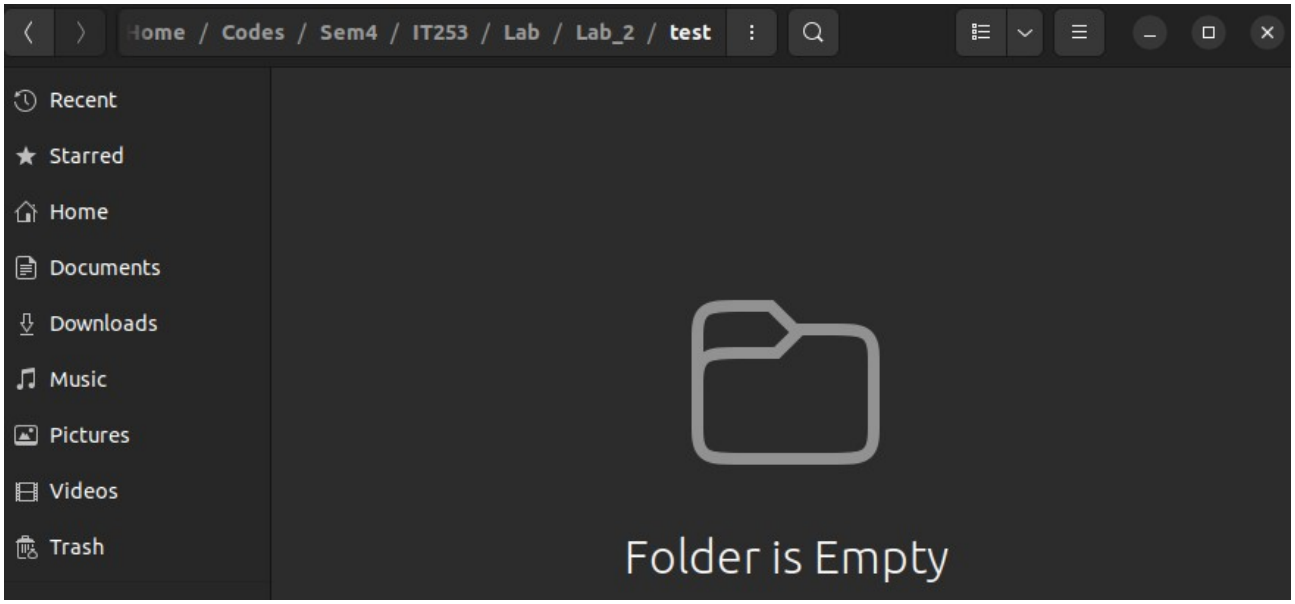


11. Write a shell script that takes a name of a folder, and delete all sub folders of size 0

```
#!/bin/sh

if [ "$#" -ne 1 ]; then
    echo "Enter valid folder path"
    exit 1
fi

folder_path="$1"

ls "$folder_path" |
    while read folder; do
        if [ -d "$folder_path/$folder" ]; then
            files=$(ls "$folder_path/$folder" | wc -l)
            if [ "$files" -eq 0 ]; then
                rmdir "$folder_path/$folder"
            fi
        fi
    done
```

OUTPUT

```
nithin@nithin1729s:~/Codes/Sem4/IT253/Lab/Lab_2$ chmod +x script11
nithin@nithin1729s:~/Codes/Sem4/IT253/Lab/Lab_2$ ./script11
Enter valid folder path
nithin@nithin1729s:~/Codes/Sem4/IT253/Lab/Lab_2$ ./script11 test
nithin@nithin1729s:~/Codes/Sem4/IT253/Lab/Lab_2$
```



12. Write a shell script that will take an input file and remove identical lines (or duplicate lines from the file)

```
#!/bin/sh

if [ "$#" -ne 1 ]; then
    echo "Enter valid file path"
    exit 1
fi

input_file="$1"
temp_file="temp_sorted.txt"

cat "$input_file" | sort | uniq >"$temp_file"
mv "$temp_file" "$input_file"
```

OUTPUT


```
nithin@nithin1729s:~/Codes/Sem4/IT253/Lab/Lab_2$ chmod +x script12
nithin@nithin1729s:~/Codes/Sem4/IT253/Lab/Lab_2$ ./script12 dups.txt
nithin@nithin1729s:~/Codes/Sem4/IT253/Lab/Lab_2$
```