Operating System (CS301)

Assignment - 2

**U19CS012**

(1) Shell script Program to accept a character and check whether it is an

* Lower case alphabet
* Upper case alphabet
* A digit
* Special symbol
* Vowel

Using case control structure.

**Script:**

*# Path to Intepretor*

*#! /bin/bash*

*# Prompt the User to Read a Character*

read -p "Enter One Character : " USER\_CHAR

*case* $USER\_CHAR *in*

    "A") echo "$USER\_CHAR is UPPER case Vowel Alphabet";;

    "E") echo "$USER\_CHAR is UPPER case Vowel Alphabet";;

    "I") echo "$USER\_CHAR is UPPER case Vowel Alphabet";;

    "O") echo "$USER\_CHAR is UPPER case Vowel Alphabet";;

    "U") echo "$USER\_CHAR is UPPER case Vowel Alphabet";;

    "a") echo "$USER\_CHAR is lower case Vowel Alphabet";;

    "e") echo "$USER\_CHAR is lower case Vowel Alphabet";;

    "i") echo "$USER\_CHAR is lower case Vowel Alphabet";;

    "o") echo "$USER\_CHAR is lower case Vowel Alphabet";;

    "u") echo "$USER\_CHAR is lower case Vowel Alphabet";;

    [[:digit:]]) echo "$USER\_CHAR is Digit";;

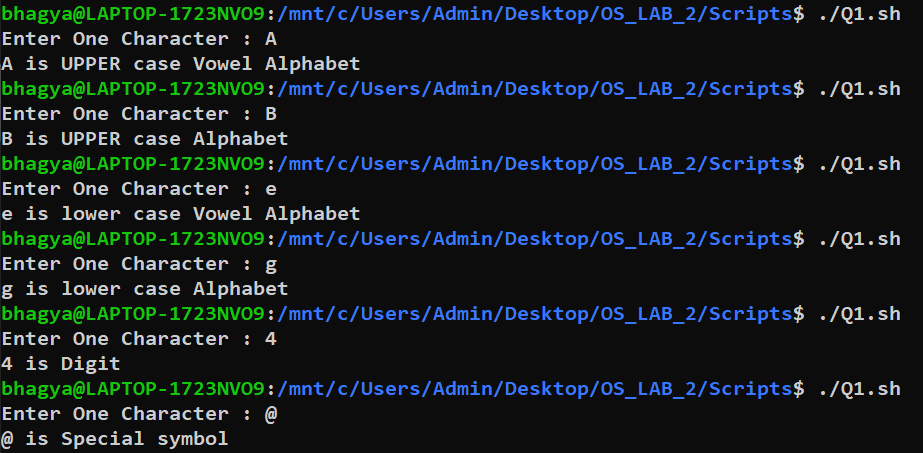
    [[:upper:]]) echo "$USER\_CHAR is UPPER case Alphabet" ;;

    [[:lower:]]) echo "$USER\_CHAR is lower case Alphabet" ;;

    \*) echo "$USER\_CHAR is Special symbol"  ;;

*esac*

**Output:**



(2) Using case .. esac structure

* Find the number of users logged into the system
* Print the calendar for current year
* Print the date

**Script:**

*#!/bin/bash*

echo "Enter n -> Number of Users Logged In"

echo "Enter c -> Calendar of current year"

echo "Enter d -> Date"

read choice

echo ""

*case* "$choice" *in*

    'n')

*# Counting unique logins when \*who\* doesn't have a --count flag.*

        users=$(who | sort --key=1,1 --unique | wc --lines)

        echo "Number of Users Logged in: " $users

    ;;

    'c')

*# cal : Shows current month calendar on the terminal with the current date highlighted.*

        cal -y

    ;;

    'd')

*# date command is used to display the system date and time*

        date

    ;;

    \*)

        echo "Please Enter a Valid Character"

    ;;

*esac*

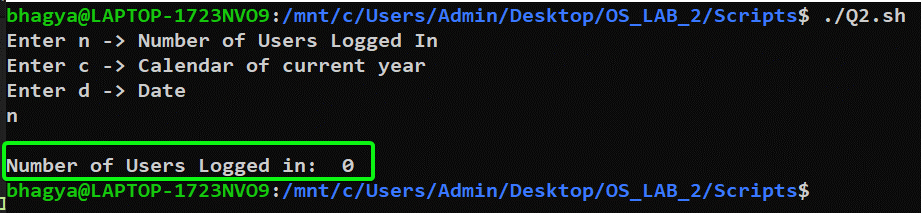
*# References:*

*# 1.) https://stackoverflow.com/questions/28208069/linux-bash-script-to-determine-number-of-users-logged-in*

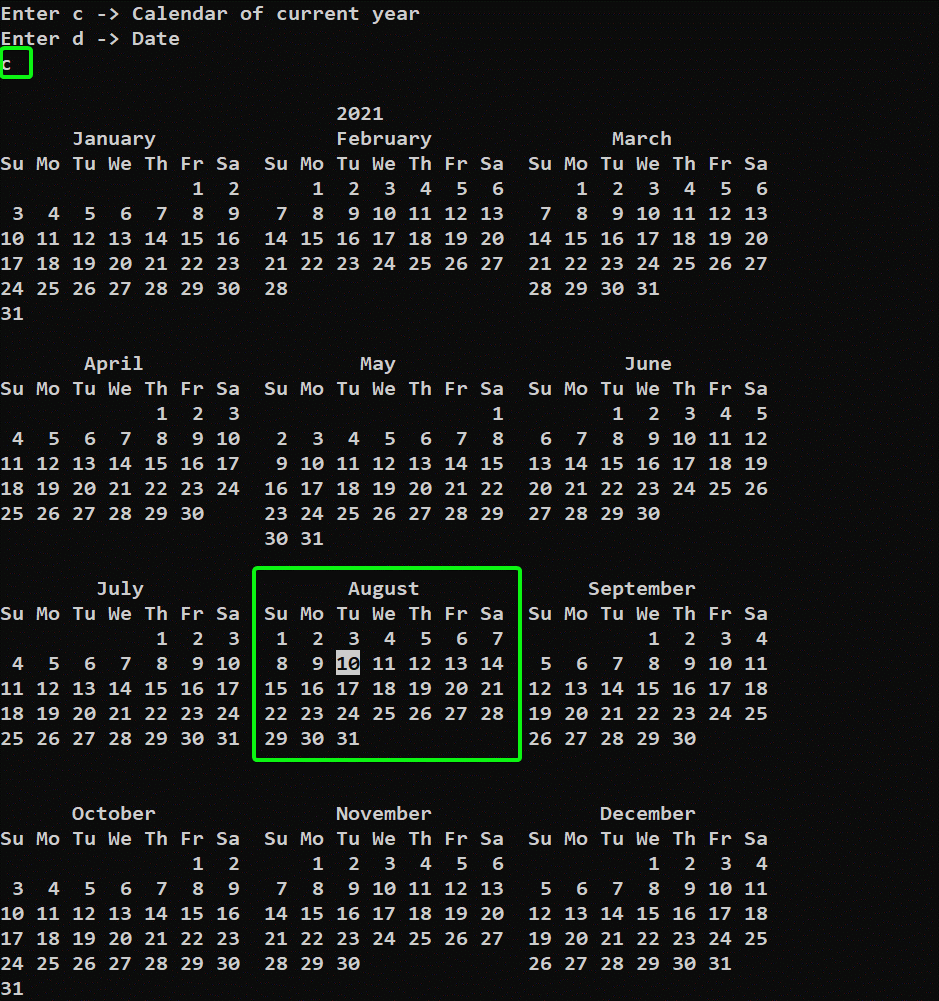
*# 2.) https://www.geeksforgeeks.org/cal-command-in-linux-with-examples/*

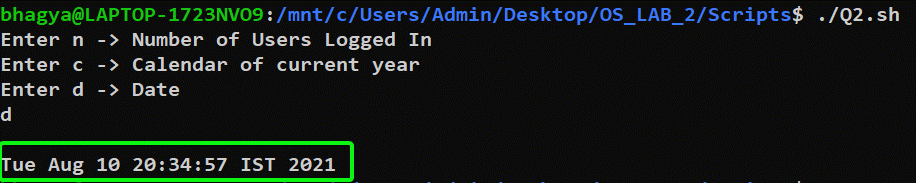
*# 3.) https://www.geeksforgeeks.org/date-command-linux-examples/*

**Output:**



Continued on Next Page!





(3) Shell Script Program to check whether given file is a directory or not.

**Script:**

*#!/bin/bash*

FILE\_NAME=$1

*if* [ -d "${FILE\_NAME}" ];

*then*

    echo "${FILE\_NAME} is a Directory"

*elif* [ -f "${FILE\_NAME}" ]; *then*

    echo "${FILE\_NAME} is a File"

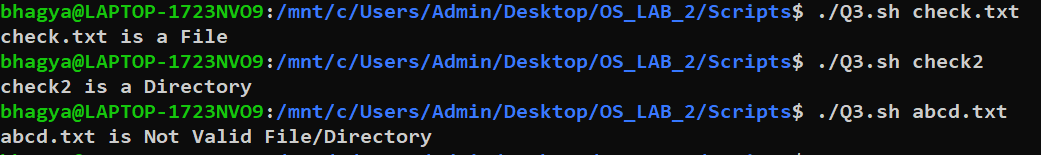
*else*

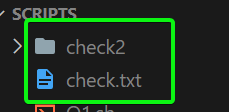
    echo "${FILE\_NAME} is Not Valid File/Directory"

    exit 1

*fi*

**Output:**





(4) Shell Script Program to Count Number of files in a Directory.

**Script:**

*#!/bin/bash*

printf "Number of Files in Current Directory: "

ls -1q | wc -l

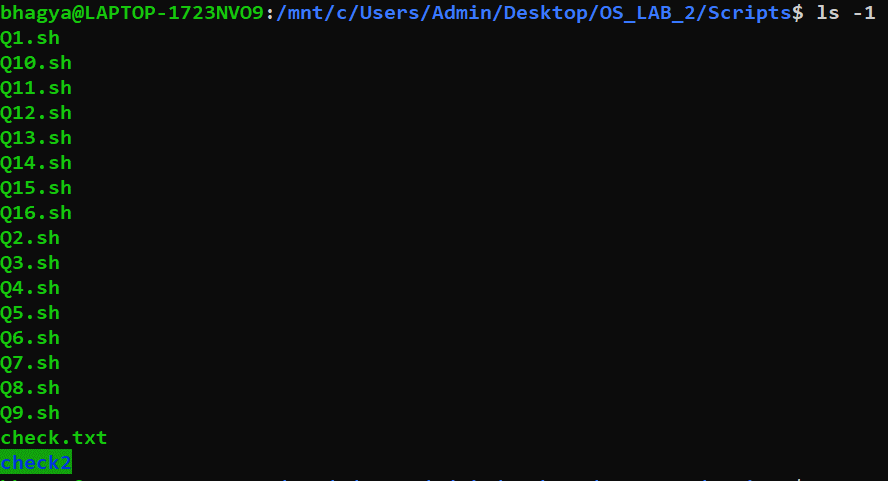
*# Reference*

*# 1.) https://stackoverflow.com/questions/20895290/count-number-of-files-within-a-directory-in-linux*

*# 2.) https://www.geeksforgeeks.org/practical-applications-ls-command-linux/*

**Output:**





(5) Shell Script Program to copy contents of one file to another.

**Script:**

*#!/bin/bash*

file1=$1

file2=$2

*if* [ -f "$file1" ]; *then*

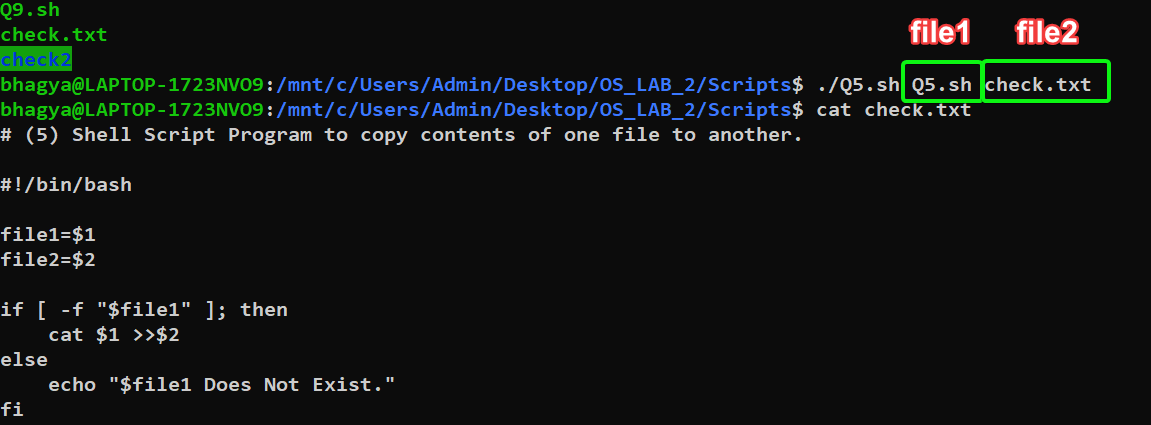
    cat $1 >>$2

*else*

    echo "$file1 Does Not Exist."

*fi*

**Output:**



(6) Write a shell script to add two numbers supplied by user and supplied as command line argument.

**Script:**

*#!/bin/bash*

*# Regular Expression for Number*

re='^[+-]?[0-9]+?$'

*# Check if Input is Valid*

*if* ! [[ $1 =~ $re ]]; *then*

    echo "Error: Input is Not a Valid Number" >&2

    exit 1

*fi*

*if* ! [[ $2 =~ $re ]]; *then*

    echo "Error: Input is Not a Valid Number" >&2

    exit 1

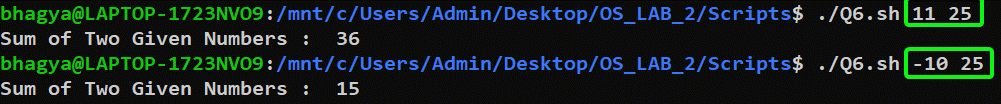
*fi*

echo "Sum of Two Given Numbers : " $(($1 + $2))

*# References*

*# 1.) https://stackoverflow.com/questions/806906/how-do-i-test-if-a-variable-is-a-number-in-bash/806923*

**Output:**



(7) Write a shell script to find out biggest number form given three numbers. Numbers are supplied by command line argument.

**Script:**

*#!/bin/bash*

*# Regular Expression for Number*

re='^[+-]?[0-9]+?$'

*# Input Validation*

*if* ! [[ $1 =~ $re ]]; *then*

    echo "Error: Input is Not a Valid Number" >&2

    exit 1

*fi*

*if* ! [[ $2 =~ $re ]]; *then*

    echo "Error: Input is Not a Valid Number" >&2

    exit 1

*fi*

*if* ! [[ $3 =~ $re ]]; *then*

    echo "Error: Input is Not a Valid Number" >&2

    exit 1

*fi*

*# Method 1*

*# Store it in Array for*

arr=($1 $2 $3)

*# Intializa the First element to be Maximum*

max\_element=${arr[0]}

*# Using ${arr[@]} or ${arr[\*]}, we can access the all array elements.*

*for* n *in* "${arr[@]}";

*do*

    ((n > max\_element)) && max\_element=$n

*done*

echo "Maximum of all 3 Numbers : " $max\_element

*# Method 2*

*if* [ $1 -gt $2 ] && [ $1 -gt $3 ]

*then*

    echo "Maximum Element : " $1

*elif* [ $2 -gt $1 ] && [ $2 -gt $3 ]

*then*

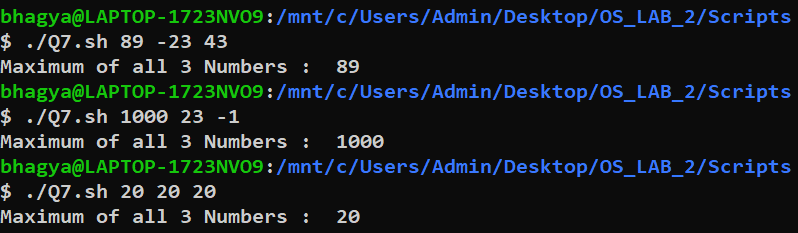
    echo "Maximum Element : " $2

*else*

    echo "Maximum Element : " $3

*fi*

**Output:**



(8) Implement simple calculator. Numbers are supplied by command line argument.

**Script:**

*#!/bin/bash*

*# Check for Valid Input*

*if* [ $# -ne 2 ]; *then*

    echo "2 command line arguments are required"

    exit 2

*fi*

*if* ! [[ $1 =~ $re ]]; *then*

    echo "error: Not a number" >&2

    exit 1

*fi*

*if* ! [[ $2 =~ $re ]]; *then*

    echo "error: Not a number" >&2

    exit 1

*fi*

*# Inputs from User*

a=$1

b=$2

echo "Enter Choice :"

echo "1. Addition"

echo "2. Subtraction"

echo "3. Multiplication"

echo "4. Division"

echo "Your Choice : "

read -n1 ch

echo " "

*# Switch Case to perform calulator operations*

*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)

    ;;

    \*)

        echo "Invalid Choice"

        exit 1

    ;;

*esac*

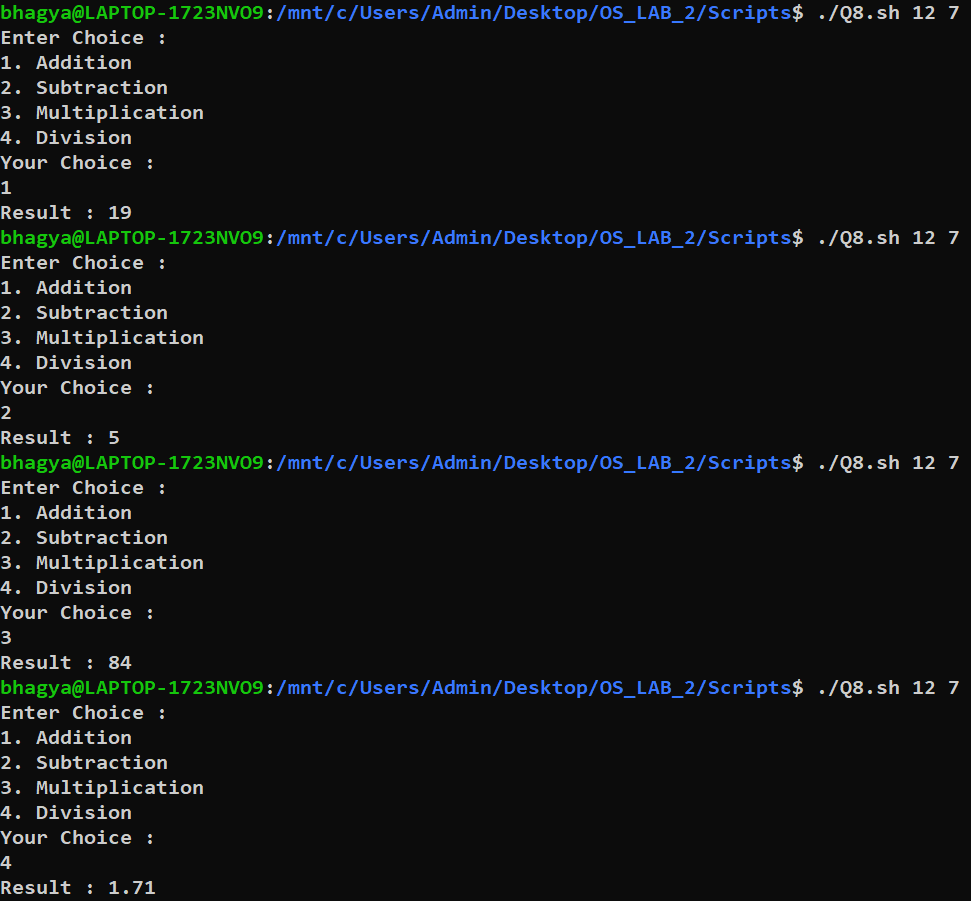
echo "Result : $res"

*# bc command is used for command line calculator.*

*# References*

*# 1.) https://www.geeksforgeeks.org/simple-calculator-bash/*

**Output:**



(9) Write a shell script to print numbers in descending order using while loop.

**Script:**

*#! /bin/bash*

*# User Need to Change this Input*

arr=(15 71 120 7 3 35)

n=${#arr[@]}

echo "Original Number(s) in Array : "

*for* x *in* "${arr[@]}";

*do*

    printf "$x "

*done*

echo ""

*for* ((i = 0; i < ${#arr[@]}; i++));

*do*

*for* ((j = 0; j < ${#arr[@]}; j++));

*do*

*if* [[ ${arr[$j]} -lt ${arr[$i]} ]]; *then*

            tmp=${arr[$i]}

            arr[$i]=${arr[$j]}

            arr[$j]=${tmp}

*fi*

*done*

*done*

echo ""

echo "Final Number(s) of Array [Descending] : "

i=0

*while* [ $i -lt $n ];

*do*

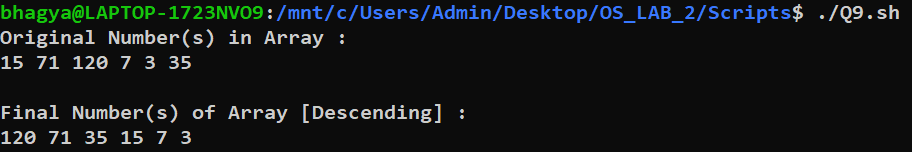
    printf "${arr[i]} "

    i=$((i + 1))

*done*

echo ""

**Output:**



(10) Write a shell script to create a simple calculator using switch-case statement.

[Same as Question 8]

**Script:**

*#!/bin/bash*

*# Check for Valid Input*

*if* [ $# -ne 2 ]; *then*

    echo "2 command line arguments are required"

    exit 2

*fi*

*if* ! [[ $1 =~ $re ]]; *then*

    echo "error: Not a number" >&2

    exit 1

*fi*

*if* ! [[ $2 =~ $re ]]; *then*

    echo "error: Not a number" >&2

    exit 1

*fi*

*# Inputs from User*

a=$1

b=$2

echo "Enter Choice :"

echo "1. Addition"

echo "2. Subtraction"

echo "3. Multiplication"

echo "4. Division"

echo "Your Choice : "

read -n1 ch

echo " "

*# Switch Case to perform calulator operations*

*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)

    ;;

    \*)

        echo "Invalid Choice"

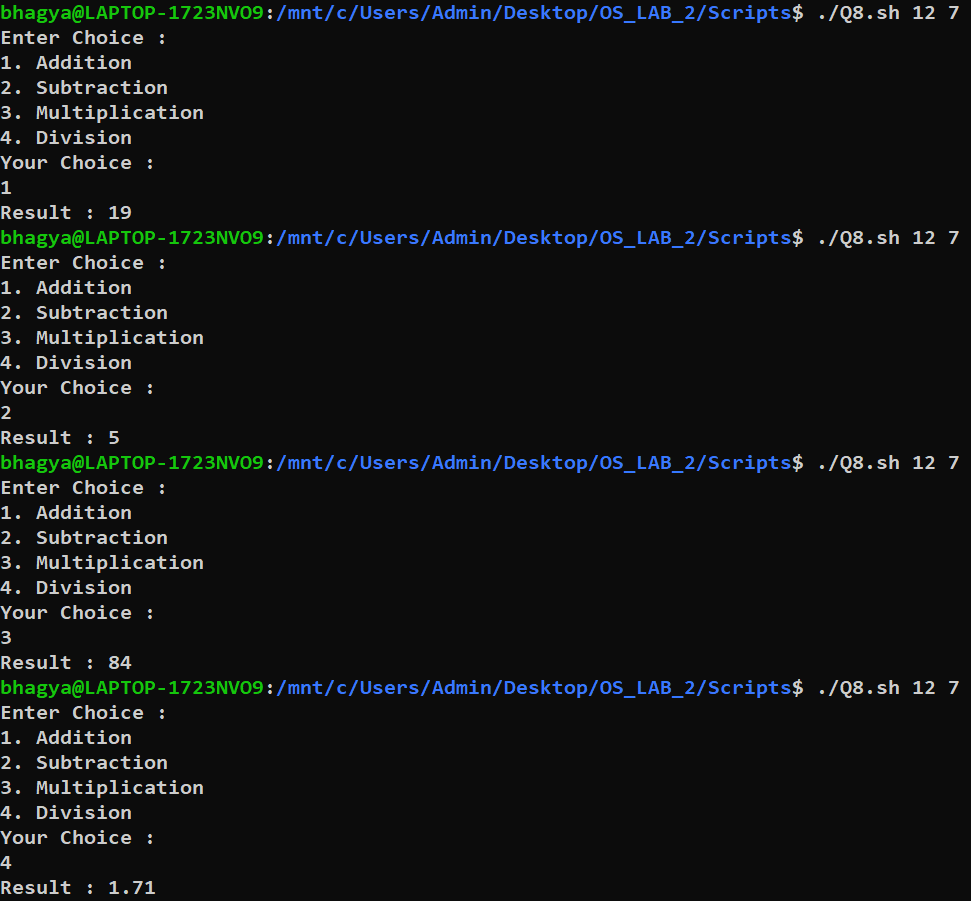
        exit 1

    ;;

*esac*

echo "Result : $res"

**Output:**



(11) Write a shell script to print given number in reverse order.

**Script:**

*#! /bin/bash*

*# Method 1*

*# Number can be Treated as String*

read -p "Enter a Number: " num

echo $num | rev

*# Method 2*

echo "Enter a Number : "

read n

digit=0

rev=0

*while* [ $n -gt 0 ]

*do*

    digit=$(( $n % 10 ))

*# Left Shift Old Answer and Add the Current Digit to it*

    rev=`expr $rev \\* 10 + $digit`

    n=$(( $n / 10 ))

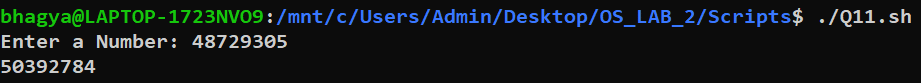
*done*

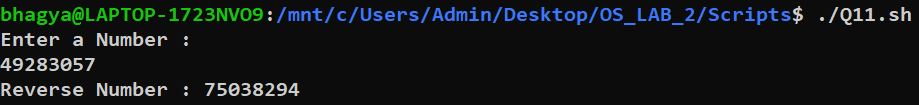
echo "Reverse Number : $rev"

*# References*

*# 1.) https://unix.stackexchange.com/questions/65913/print-given-number-in-reverse-order/65924*

**Output:**





(12) Write a shell script to print sum of all digits of a given number.

**Script:**

*#!/bin/bash*

read -p "Enter Number : " num

*# Intialize the Sum = 0*

sum=0

*while* [ $num -gt 0 ];

*do*

    digit=$((num % 10))

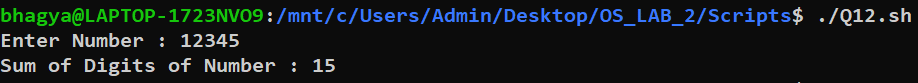
    sum=$((sum + digit))

    num=$((num / 10))

*done*

echo "Sum of Digits of Number : $sum"

**Output:**



(13) Find the factorial value of given input number.

**Script:**

*#! /bin/bash*

read -p "Enter Number: " num

re='[0-9]'

*if* ! [[ $num =~ $re ]]; *then*

    echo "Error : Not A Valid Number for Factorial" >&2

    exit 1

*fi*

fact=1

*while* [ $num -gt 1 ]; *do*

    fact=$((fact \* num))

    num=$((num - 1))

*done*

echo "Factorial of Given Number is : $fact"

*# Limitation : It can Only Calculate Small Factorial*

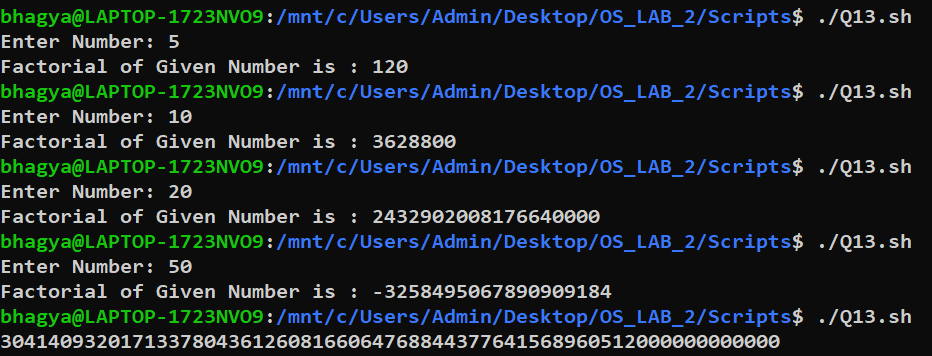
*# For Solving the Limitation :*

echo 'define f(x) {if (x>1){return x\*f(x-1)};return 1} f(100)' | bc

*# References*

*# 1.) https://stackoverflow.com/questions/3394580/how-do-you-find-the-factorial-of-a-number-in-a-bash-script*

**Output:**



(14) Generate and display Fibonacci series.

**Script:**

*#! /bin/bash*

read -p "Enter number of values to show in Fibonnaci series: " N

echo "The Fibonacci series is : "

a=0

b=1

*for* ((i = 0; i < N; i++));

*do*

    echo -n "$a "

    fn=$((a + b))

    a=$b

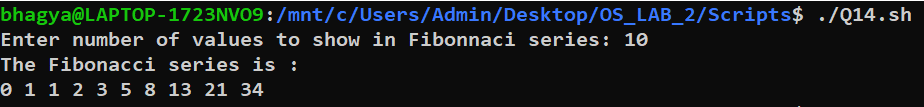
    b=$fn

*done*

echo ""

*# echo -n : this option is used to omit echoing trailing newline .*

**Output:**



(15) Display all even numbers within given range.

**Script:**

*#! /bin/bash*

read -p "Enter Lower Limit of Range : " first

read -p "Enter Upper Limit of Range : " second

*# Approach 1 : Brute Force*

*for* ((i = $first; i <= $second; ++i));

*do*

    rem=$((i % 2))

*if* [ "$rem" -eq "0" ]; *then*

        echo $i

*fi*

*done*

*# Approach 2 : Since Modulus is Costly Operation*

rem2=$((first % 2))

*if* [ "$rem2" -eq "1" ]; *then*

    ((first++))

*fi*

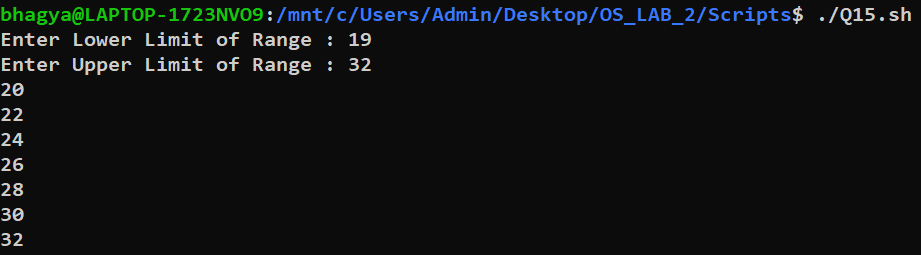
*for* ((i = $first; i <= $second; i+=2));

*do*

    echo $i

*done*

**Output:**



(16) Find out number of characters, words and lines from a given file.

**Script:**

*#! /bin/bash*

*# Using wc command the number of words, characters and number of lines can be determined.*

*if* [ -f "$1" ]; *then*

    words=$(cat $1 | wc -w)

    chars=$(cat $1 | wc -c)

    line=$(cat $1 | wc --lines)

    echo Number of Characters *in* $1 is $chars

    echo Number of Words *in* $1 is $words

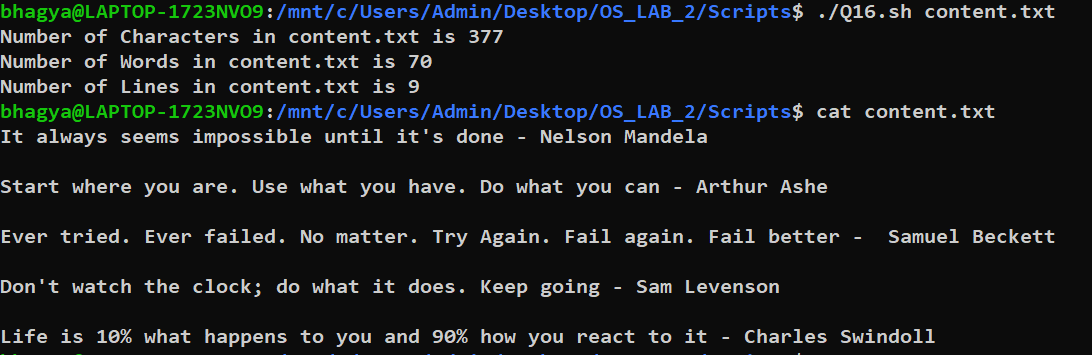
    echo Number of Lines *in* $1 is $line

*else*

    echo "Error : $1 File Does Not Exist!"

*fi*

**Output:**



SUBMITTED BY:

**U19CS012**

BHAGYA VINOD RANA