Operating System (CS301)

Assignment - 3

**U19CS012**

1) Write a shell script, which finds the prime factors of a given number.

**Script:**

read -p "Enter Number whose Prime Factors Needs to be Found : " no

*# Invalid Input [Negative Numbers]*

*if* [ $num -lt 1 ]; *then*

    echo "Only Positive Numbers Allowed!"

    exit 1

*fi*

*# Special Case of 1*

*if* [ $num -eq 1]; *then*

    echo "1 is Unit. It has No Prime Factors!"

    exit 1

*fi*

echo "The Prime Factors of $no are:"

*for* ((i = 2; i <= $no; i++));

*do*

*while* [ $((no % $i)) == 0 ];

*do*

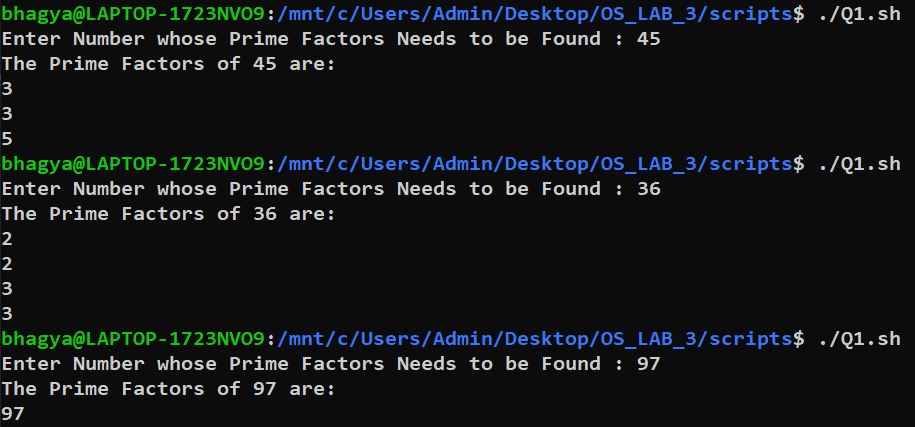
        echo $i

        no=$((no / $i))

*done*

*done*

**Output:**



2) Write a shell script that accepts a positive integer value from the user, say 34, and prints out all the divisors of 34 as a list:

Enter a positive integer: 34

The divisors of 34 are: 1, 2, 17, and 34

**Script:**

read -p "Enter A Number whose Divisors need to be Found : " num

*# Only Positive Numbers Allowed*

*if* [ $num -lt 1 ]; *then*

    echo "Only Positive Numbers Allowed!"

    exit 1

*fi*

printf "The divisiors of $num are: "

*for* ((i=1; i <= $num; i++)); *do*

*if* [ $(expr $num % $i) -eq 0 ]; *then*

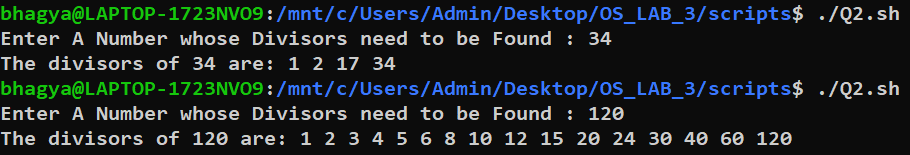
        echo -n "$i "

*fi*

*done*

echo ""

**Output:**



3) Write a shell script, which prints good morning or good evening depending on the login time of the user.

**Script:**

hour=$(date +%H)

*if* [ $hour -lt 12 ]; *then*

    echo "Good Morning"

*elif* [ $hour -lt 16 ]; *then*

    echo "Good Afternoon"

*elif* [ $hour -lt 20 ]; *then*

    echo "Good Evening"

*else*

    echo "Good Night"

*fi*

**Output:**



4) A shell script, which takes as command line input a number n, and a word. It then prints the word n times, once on each line.

**Script:**

*# $# -> No of Argument [Count]*

*if* [ $# -ne 2 ]; *then*

    echo "2 Command Line Arguments Required"

    exit 1

*fi*

*# First Argument: Word*

word=$1

*# Second Argument: No of Times to be Printed*

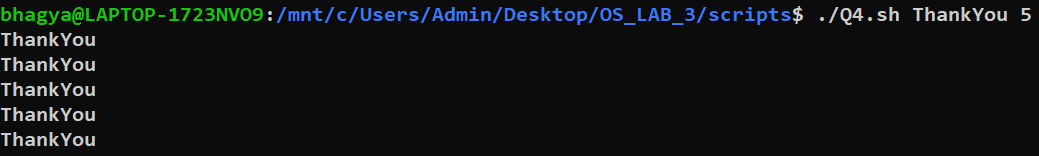
num=$2

*for* ((i=0; i < $num;i++ )); *do*

    echo "$word"

*done*

**Output:**



5) Write a shell script, which finds the total number of blank lines in the given file.

**Script:**

*#!/bin/bash*

read -p "Enter a File name: " file

echo "Number of blank space in $file is: "

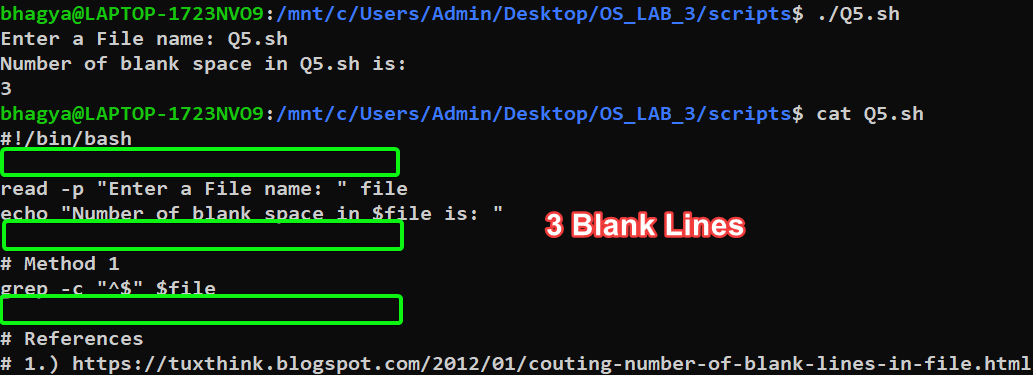
*# Method 1*

grep -c "^$" $file

*# References*

*# 1.) https://tuxthink.blogspot.com/2012/01/couting-number-of-blank-lines-in-file.html*

**Output:**



6) A shell script, which reports the names and sizes of all the files in a directory whose size exceeds 1000 bytes, in descending order of their sizes and the total number of such files.

**Script:**

echo "Files greater than 1000 bytes : "

find . -maxdepth 1 -type f -size +1000c -ls | sort -r -n -k7

echo "Number of such files : "

find . -maxdepth 1 -type f -size +1000c -ls | sort -r -n -k7 | wc -l

*# -maxdepth 1 : find files only in current directory*

*# -type f : find only files*

*# -size +1000c : find only files greather than 1000 bytes ("c" = bytes)*

*# -ls : list them*

*# -r Option: Sorting In Reverse Order*

*# -n Option : To sort a file numerically*

*# -k7 Option : to sort on the seventh column.*

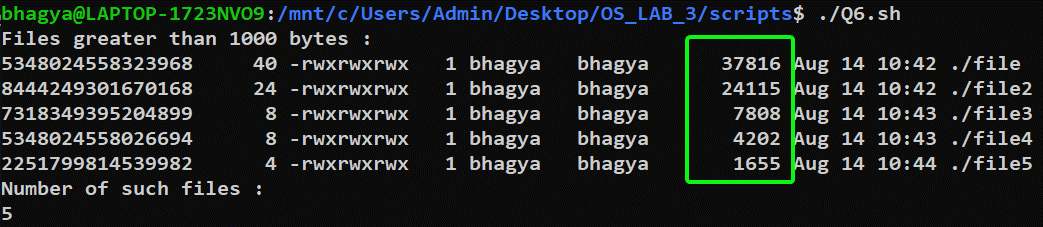
*# wc -l : To Count the Number of Lines*

*# References*

*# 1.) https://unix.stackexchange.com/questions/394672/sort-files-greater-than-1000-bytes-in-descending-order*

*# 2.) https://www.geeksforgeeks.org/sort-command-linuxunix-examples/*

**Output:**



SUBMITTED BY:

**U19CS012**

BHAGYA VINOD RANA