# OS LAB ASSIGNMENT 1

Siddhanth Gupta
BCSE-III ROLL: 001210501021

# CONTENTS

# PROGRAM 1

## *Part A*

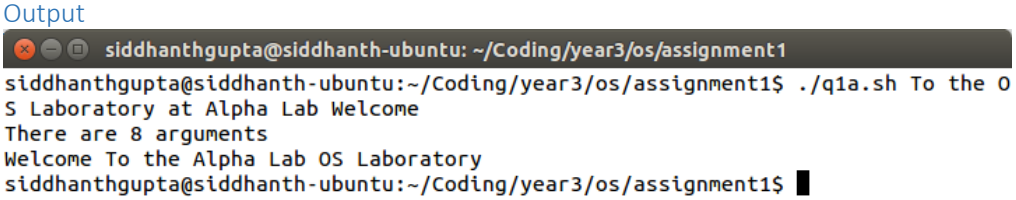
### Problem Statement

Write a shell script that reads command line arguments and prints the number of arguments. Given this sequence of words as command line arguments “To the OS Laboratory at Alpha Lab Welcome”, the script should echo back the information that there are 8 arguments. Additionally, it should echo back some of the arguments in the following order: “Welcome To the Alpha Lab OS Laboratory”.

### Source Code

```
#!/bin/bash

#This is displaying the number of command line arguments
echo "There are $# arguments"

#Doing this, 'To the OS Laboratory at Alpha Lab Welcome' becomes
#            'Welcome to the Alpha Lab OS Laboratory'
echo "$8 $1 $2 $6 $7 $3 $4"
```

### Output




## *Part B*

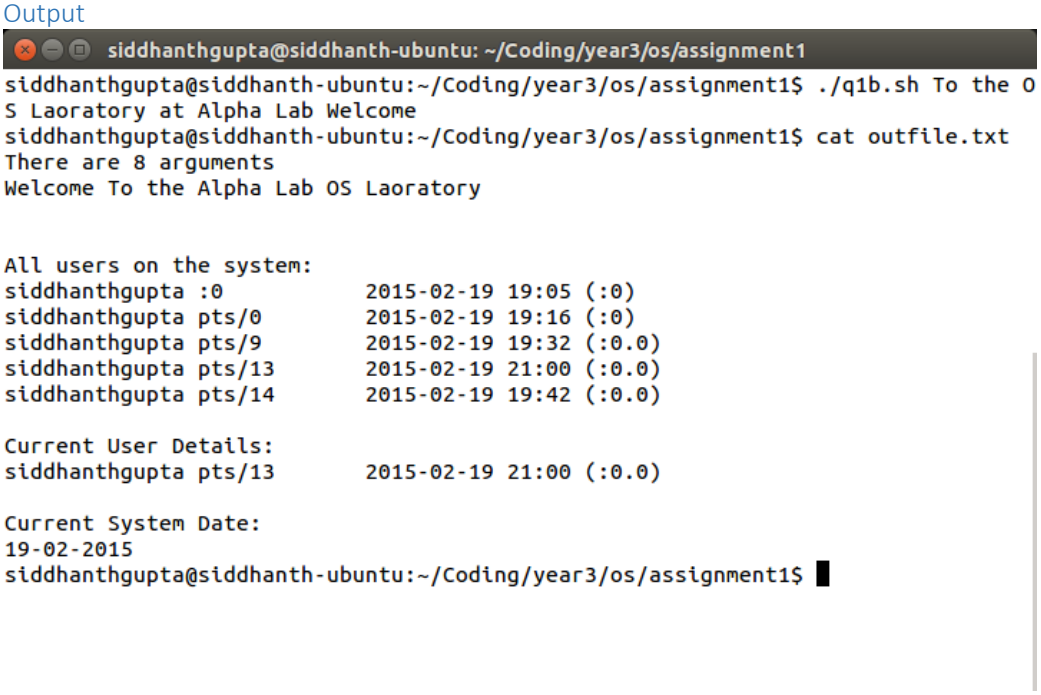
## Problem Statement

Redirect the displayed line to a file. Append the outputs of the commands to show the users logged into your system, you login account and the current date in the dd-mm-yyyy format to the file

## Source Code

```
#!/bin/bash

#Deletes outfile.txt if it already exists
if [ -e "outfile.txt" ];then
    rm outfile.txt
fi

touch outfile.txt
#Creates the outfile.txt

echo "There are $# arguments" >> outfile.txt
#Number of arguments passed over command line
#output written to file

echo "$8 $1 $2 $6 $7 $3 $4" >> outfile.txt
#Rearranged argument written to file
echo >> outfile.txt
#One line whitespace to neaten the output

echo -e "\nAll users on the system: " >> outfile.txt
who >> outfile.txt
#The different users logged onto the current system

echo -e "\nCurrent User Details: " >> outfile.txt
who am i >> outfile.txt
#Complete details of the current user

echo -e "\nCurrent System Date: " >> outfile.txt
date "+%d-%m-%Y">> outfile.txt
#Current System date
```

## Output


siddhanthgupta@siddhanth-ubuntu: ~/Coding/year3/os/assignment1
siddhanthgupta@siddhanth-ubuntu:~/Coding/year3/os/assignment1$ ./q1b.sh To the O
S Laoratory at Alpha Lab Welcome
siddhanthgupta@siddhanth-ubuntu:~/Coding/year3/os/assignment1$ cat outfile.txt
There are 8 arguments
Welcome To the Alpha Lab OS Laoratory
All users on the system:
siddhanthgupta :0 2015-02-19 19:05 (:0)
siddhanthgupta pts/0 2015-02-19 19:16 (:0)
siddhanthgupta pts/9 2015-02-19 19:32 (:0.0)
siddhanthgupta pts/13 2015-02-19 21:00 (:0.0)
siddhanthgupta pts/14 2015-02-19 19:42 (:0.0)
Current User Details:
siddhanthgupta pts/13 2015-02-19 21:00 (:0.0)
Current System Date:
19-02-2015
siddhanthgupta@siddhanth-ubuntu:~/Coding/year3/os/assignment1$

# PROGRAM 2

## Problem Statement

Create a Directory using your roll number (CSE-xx) as its name

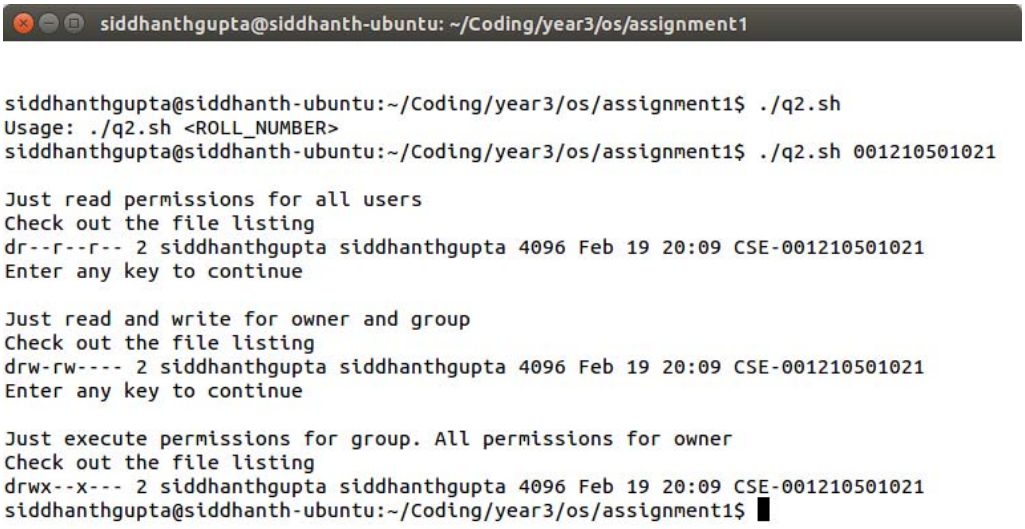
a) Allow everyone to list files in CSE-xx. No other privileges are to be changed
b) Allow the owner and group members to list, remove or add files in CSE-xx. All privileges are to be removed from all others
c) Allow the owner and group members to execute CSE-xx. Only the owner will have read and write permission.

## Source Code

```
#!/bin/bash

#If there are no command line arguments, shows an error and exits
if [ "$#" -eq 0 ]; then
    echo "Usage: ./q2.sh <ROLL_NUMBER>"
    exit 1
fi

#If the directory exists, shows appropriate message
#else, creates the new directory
if [ -d "CSE-$1" ]; then
    echo "Directory Exists. Not creating again"
else
    mkdir "CSE-$1"
fi
declare dir_name="CSE-$1"

#Q2(a)
#Allows READ permissions for user, group and others
chmod 444 $dir_name
echo -e "\nJust read permissions for all users"
echo "Check out the file listing"
ls -al | grep "$dir_name"
read -p "Enter any key to continue" yn


#Q2(b)
#Allows READ and WRITE permissions for user and group
#No permissions for others
chmod 660 $dir_name
echo -e "\nJust read and write for owner and group"
echo "Check out the file listing"
ls -al | grep "$dir_name"
read -p "Enter any key to continue" yn


#Q2(c)
#Allows READ, WRITE, EXEC permissions for the user
#only EXEC permission for group. No permissions for others
chmod 710 $dir_name
echo -e "\nJust execute permissions for group. All permissions for
owner"
echo "Check out the file listing"
```

```
ls -al | grep "$dir_name"
```

Output



# PROGRAM 3

## Problem Statement

Write a shell script that

a) Counts the total number of files/sub-directories in CSE-xx
b) Lists the number of files and number of subdirectories in CSE-xx. All privileges are to be removed from all others
c) Repeat steps (a) and (b) for each subdirectory of CSE-xx

## Source Code

```
#!/bin/bash

#The function foo() lists the number of files/directories in the folder
#and also lists the details of files/directories inside the folder
function foo()
{

      declare x=$(ls | wc -l)
      #Returns the linecount of the ls command. Gives the number of

      #files/directories in the present working directory

      declare dir_count=$(ls -l | grep "^d"|  wc -l)
      #Returns the number of directories in the present working dir
      declare file_count=$(($x-$dir_count))
      #Diff between the previous two results gives the number of files

      printf "%d FILES, %d FOLDERS\n" $file_count $dir_count

}


tab_value=0
#Since output is like a tree, tab value keeps tracks of the tabs
#at the beginning of a line of the output

function fooRecur()
#Calls the function foo for the current directory
    for dir in */
      do
        printf "\t%.0s" $tab_value
        printf "%s: " $dir
        cd "$dir"                                  #Changes PWD
        foo
        (( tab_value+= 1))
            #Increases tab value every time it goes deeper into
            directory tree

        declare dir_count=$(ls -l | grep "^d"|  wc -l)
        if [ $dir_count -ne 0 ]; then
            #If the directory has further subdirectories, then
            fooRecur
        fi
```

```
        cd ..                                   #Changes back PWD
        (( tab_value-= 1))
    done


#Main function
echo
printf "%s: " ${PWD##*/}
foo
fooRecur
```

Output

```
siddhanthgupta@siddhanth-ubuntu: ~/Coding/year3/os/assignment1
siddhanthgupta@siddhanth-ubuntu:~/Coding/year3/os/assignment1$ ./q3.sh

assignment1: 13 FILES, 3 FOLDERS
        CSE-001210501021/: 0 FILES, 0 FOLDERS
        my_shell/: 2 FILES, 0 FOLDERS
        test_fol/: 4 FILES, 0 FOLDERS
siddhanthgupta@siddhanth-ubuntu:~/Coding/year3/os/assignment1$
```

# PROGRAM 4

## Problem Statement

Write a shell script to count the number of lines in any file.

a) Name of the file is to be given by the user
b) Name of the file is a command line argument

## Source Code

```
#!/bin/bash

#If filename is not passed through command-line, it asks for user input
#else, it takes the filename as the command line argument
declare fn
if [ $# -eq 0 ]; then
    echo "Command line argument not detected. Switching to manual
input"
    read -p "Enter filename: " fn
else
    echo "Read filename from command line arguments"
    fn=$1
fi


#if filename does not exist, then it displays error message
#else, displays the line count of the file
if [ ! -f $fn ]; then
    echo "File does not exist"
else
    declare x=$(cat "$fn" | wc -l)          #Counts the number of lines
in the file
    echo "File has $x lines"
fi
```

## Output

```
siddhanthgupta@siddhanth-ubuntu: ~/Coding/year3/os/assignment1
siddhanthgupta@siddhanth-ubuntu:~/Coding/year3/os/assignment1$ ./q4.sh q2.sh
Read filename from command line arguments
File has 44 lines
siddhanthgupta@siddhanth-ubuntu:~/Coding/year3/os/assignment1$
siddhanthgupta@siddhanth-ubuntu:~/Coding/year3/os/assignment1$
siddhanthgupta@siddhanth-ubuntu:~/Coding/year3/os/assignment1$ ./q4.sh
Command line argument not detected. Switching to manual input
Enter filename: q2.sh
File has 44 lines
siddhanthgupta@siddhanth-ubuntu:~/Coding/year3/os/assignment1$
```

# PROGRAM 5

## Problem Statement

Write a shell program that finds the factorial of an integer and displays the result with a suitable message. Also, mention the time taken to fund out the corresponding factorial. Test the program for different integers.

## Source Code

```
#!/bin/bash

declare STARTTIME=$(date "+%s")
declare -i count                                    #Stores loop counter
declare -i number                                   #Stores the user input
read -p "Enter number: " number                     #Accepts input from
user
count=number
declare factorial=1                                 #Stores the factorial
while [ $count -gt 0 ]                              #while counter > 0
do
   factorial=$(( $factorial * $count ))
                        #Mathematical evaluation of factorial*count
   count=$(( $count - 1 ))                  #Decrement of count variable
done

echo "$number! = $factorial"
declare ENDTIME=$(date "+%s")
echo "Time taken to complete is $(( $STARTTIME- $ENDTIME)) seconds"
```

## Output

```
siddhanthgupta@siddhanth-ubuntu: ~/Coding/year3/os/assignment1
siddhanthgupta@siddhanth-ubuntu:~/Coding/year3/os/assignment1$ time ./q5.sh
Enter number: 8
8! = 40320
Time taken to compute this is 2 seconds

real    0m1.878s
user    0m0.005s
sys     0m0.007s
siddhanthgupta@siddhanth-ubuntu:~/Coding/year3/os/assignment1$ time ./q5.sh
Enter number: 12
12! = 479001600
Time taken to compute this is 1 seconds

real    0m1.315s
user    0m0.000s
sys     0m0.011s
siddhanthgupta@siddhanth-ubuntu:~/Coding/year3/os/assignment1$ time ./q5.sh
Enter number: 0
0! = 1
Time taken to compute this is 1 seconds

real    0m1.255s
user    0m0.002s
sys     0m0.005s
siddhanthgupta@siddhanth-ubuntu:~/Coding/year3/os/assignment1$
```