**SCRIPTING ASSIGNMENTS**

## Day 1 Assignments:

1. Create a Environemental variable EDITOR in your shell **at the command line** with any editor of your choice ( for eg vim / notepad/ gedit etc..) . Do the following:
   1. Create a file hello.txt by invoking EDITOR command on the shell . Write any contents on the file 1.txt and save and quit.
   2. Display on the command line the file you have created. Now exit the shell.
   3. Open another shell. Repeat the steps a) & b). What is your obseravtion does the command EDITOR work in the new shell? Why it doesn't work?
2. The task is to make the command EDITOR work permanently. Put it in an appropriate file so that whenever you login, the EDITOR environmental variable is availabe for you.
3. Write a script to take the full path of the file 1.txt. Display the contents of the file if it exists and is of non-zero size. If the file is not found, error message should be displayed.

Store only the filename in the variable FILENAME. Display its value.

1. The variable MYVAR=foodforthought.jpg. What would be the output :
   * + 1. echo ${MYVAR##\*fo}
       2. echo ${MYVAR#\*fo}
2. The Variable MYFOOD="chickensoup". What is displayed when the command:

echo ${MYFOOD%%soup} is executed.

1. Create aliases for the commands :
   1. ls -l as 'listlong'
   2. clear as 'c'
2. Whats the output of the below?

echo -e “Whats ur age ? \c Please enter your age'

Write why its so.

1. Write a script to print the number of arguments But print only the arguments which are not skipped. Read as input the skip argument. You can assume the skip number is less than the total number of arguments.

For eg. If 1 2 3 4 'a b c' 5 6 are the arguments, and the skip number is 2, it should output only 1 4 6

1. Write the below lines in a script as here document and execute the script.

I am so and so

I am using $SHELL as shell

1. Set the PATH variable to search in your current working directory temporarily.
2. What will be the output of the below command?

printf '\c%b' '\t hello'

1. Find the number of lines in list.words?
2. Write a Regex to match the word morning in the text 'Hello good morning' .
3. From the grep man page find the switches for the following:
   1. Ignore case in input:
   2. Print lines that don't match
   3. List matching input files

Construct your pattern and use grep for the above on any text file of your choice.

1. What does the below regexes match?
   1. echo 123 you 123 | grep --colour '[^[:alpha:][:space:]]\{2\}'
   2. echo 1a3 1ab3 1aax3 18 1 | grep --color '1[^[:alpha:]]\*3' # No matches
   3. echo pre ab123xx321xac post | grep --color '\(a[bc]\)\([0-9][0-9]\*\)xx\2x\1'
   4. echo ac abc abbc abbbbc | egrep --color 'ab+c' #matches abc abbc abbbbc

## Day 2 Assignments:

1. Write a function to take the name of a file as an argument. The script should display what type of file it is. For eg.

$ scriptname thisfile.tar

is given as input to the script, the script should display its a tar file.

The script should be able to detect files of type .tar, .jpg, .zip, .txt files

1. Write a script to take the full paths of files as arguments. Write functions:
   * + - 1. Function name display\_funcion() : to display the contents of the file if it exists else return error
         2. Write a function to split the filename into directory & filename .
2. Write a script to print out only the users who are logged into the system . There can be many users logged multiple times into the system. For eg. Users Arvind , Govind, Sita have all logged in multiple times. The output should only print the users in alphabetical order with each user appearing once. Use awk program to find only everything except the user in the output of who command.
3. Write a script to create a user testuser. Mkdir directories 1 , 2 inside testuser. The script should be able to detect the directories only and create a back-up directory by name

testuser-backup, testuser-backup/1 , testuser-backup/2. You can use the sed program if you want.

1. Write a cmd to select only the filesystems which have xfs as its filesystem. But print only the 1st field. The input file is /etc/fstab
2. Use the sort command to sort the /etc/passwd file to sort first by group id and then by user id.

7) Selects only the filesystems having xfs in /etc/fstab

awk -F ' ' '/xfs/ { print $1 }' fstab

8) Use the sort command to sort the /etc/passwd file to sort first by group id and then by user id.

sort -t: -k4,4n -k3,3n passwd

9) Find the ip address of a system from the info presented by issuing the command ifconfig -a .

Display only the ip address.

ifconfig -a | awk '/cast/ { print $2 }'

10) Open a Terminal. Export a variable Xvar=something. Now open a new terminal by opening a new Tab and see whether the variable Xvar exists?

Why so? Now create a child bash shell within the terminal. Now see whether the variable Xvar exists or not?

What can you conclude?

Terminal created by New Tab is not a child process of the previous terminal program?

11) You run the script:

#!/bin/bash

if cd /tmp/x

then

:

else

echo could not change to /tmp/x

exit 1

fi

What is the exit status of the script?