Lab3:

The main idea of this assignment is to provide hands on experience on the following topics

Grep

Pipe

redirection

read

command Line arguments

head,tail,tr

1. Create a file “poem.txt” with the following lines

We have not wings, we cannot soar;  
But we have feet to scale and climb  
By slow degrees, by more and more,  
The cloudy summits of our time.

The mighty pyramids of stone  
That wedge-like cleave the desert airs,  
When nearer seen and better known,  
Are but gigantic flights of stairs.

The distant mountains, that uprear  
Their solid bastions of the skies,  
Are crossed by pathways that appear  
As we to higher levels rise.

The heights by great men reached and kept  
Were not attained by sudden flight,  
But they, while their companions slept,  
Were toiling upward in the night.

END

Henry Wadsworth Longfellow (1807–1882) was an American poet and educator.

Do the following task using grep command

1. Print all the lines with the pattern “they”

=> grep "they" poem.txt

1. Print all the lines other than pattern “They”

=> grep -v "They" filename

1. Print all the lines starts with “w”

=> grep “^w” poem.txt

1. Print the next lines after the pattern “stone” matches.

=> grep -A 3 "stone" filename

1. Print the 2 lines above the pattern “stone” matches

=> grep -B 2 "stone" poem.txt

1. Search the pattern with exact match

=>grep -x "stone" example.txt

1. Explore variations of grep command
2. ngrep
3. pgrep
4. zgrep
5. egrep

=> **ngrep:** Used to debug plain text protocols interactions like HTTP, SMTP, FTP, DNS, among others, or to search for a specific string or pattern, using a grep regular expression syntax.

=> **pgrep:** searches for processes based on their name and other attributes, and returns their PIDs.

=> **zgrep**: Used to search out expressions from a given a file even if it is compressed. All the options that applies to the **grep** command also applies to the **zgrep** command.

=> **egrep**: Used to scan a specified file line by line, returns lines that contain a pattern matching a given regular expression.

1. Write a shell script to get the pattern and filenames from the user and check whether the pattern is present or not.

=>findPattern.sh

read -p "Enter the pattern " pat

read -p "Enter the filename " filen

grep $pat $filen

In the terminal run, bash findPattern.sh

1. Rewrite the above shell script using command line arguments. ( pass the pattern and file through command line arguments)

=> finPattern1.sh

#!/bin/bash

pat="$1"

filen="$2"

grep "$pat" "$filen"

In the terminal run, bash filePattern1.sh “stone” poem.txt

1. Write a shell script to count total number of regular files in the current working directory.

=>getCount.sh

#!/usr/bin/bash

file\_count=$(ls -p | grep -v / | wc -l)

echo $file\_count

In terminal, run bash getCount.sh

1. pipe
2. Pick the line from 3 to 5.

=> head -n 5 poem.txt | tail -n 3

1. List the top 5 largest files in a directory and display their size

=> ls -lS | head -5

1. Print the last 2 modified file details

=> ls -lt | tail -2

1. Redirection
2. Convert uppercase into lowercase characters

=> echo 'HELLO' tr '[:upper:]' '[:lower:]'

1. List the contents of your current directory, including the ownership and permissions, and redirect the output to a file called contents.txt within your home directory.

=> tr 'A-Z' 'a-z' < input.txt > output.txt

Uppercase letters of input.txt are converted to lowercase and stored in output.txt

1. Rewrite the shell script (3) using <<

* cat << EOF

$(cat findPattern.sh)

EOF