ASSIGNMENT 1

Q1] SIMPLE

Q2,3] CREATING DIRECTORIES

FOR NESTED DIRECTORIES mkdir -p lab1/lab2/lab3

touch lab1/lab2/lab3/file{1..10}

FOR SIMPLE DIRECTORIES

mkdir labb1 labb2 labb3

touch labb3/file{1..10} (if you didnt in that specific directorie)

touch file{1..10} (if we are inside that directorie)

Q4] WHO AND WHOAMI

who = Shows all users currently logged into the system

whoami = Shows the username of the current user executing the command

Q5] Process

PS to list down process

kill process id to kill that particular process

sleep 200& to create new processs for 200 second

Q6] Create 50 files using single command

touch labb2/ data{1..50}

Q7] Copy all `.txt` files from one folder to another using wildcards.

mkdir source\_folder destination\_folder (Created two separate directories)

touch source\_folder/file{1..5}.txt (Created 5 .txt files inside)

ls source\_folder (Listed the contents of source\_folder)

cp source\_folder/\*.txt destination\_folder/ (Copied all .txt files to destination\_folder using a wildcard)

ls destination\_folder (Verified the copy by listing destination\_folder)

if we want to copy text only

echo "helllo how are you"> file1000.txt

touch file200.txt

cat file1000.txt > file200.txt

cat file200.txt

Q8] Use `cat`, `less`, `head`, `tail` to view file contents.

cat filename.txt (View Entire File)

less filename.txt (Scroll Through File)

head filename.txt (View First Few Lines)

tail filename.txt (View Last Few Lines)

Q9] Write a small C program and compile it using a `Makefile`.

make one directory go in the directorie

create nano main.c

include <stdio.h>

int main() {

printf("Hello, world!\n");

return 0;

}

then create nano makefile

CC = gcc

CFLAGS = -Wall

TARGET = hello

all: $(TARGET)

$(TARGET): main.o

$(CC) $(CFLAGS) -o $(TARGET) main.o

main.o: main.c

$(CC) $(CFLAGS) -c main.c

clean:

rm -f \*.o $(TARGET)

make

./hello

Q10] Redirect the output of `ls -l` to a file and append the date to it.

create directory mkdir redirect\_prac cd redirect\_prac

touch file1.txt file2.txt file3.txt Create Some Empty Files

ls -l > output.txt Runs ls -l (which lists files with details) and saves the result into a file called output.txt.

date >> output.txt (Adds today’s date and time to the end of output.txt.)

cat output.txt

--------------------------------------------------------------------------------------------------------

ASSIGNMENT 2

Q1] Create a file and use `chmod` to set it to `rwx`

touch minefile.txt

chmod 777 minefile.txt

ls -l minefile.txt

Q2] Create another file and change its owner using `chown`

touch anotherfile.txt

udo chown kartik anotherfile.txt

ls -l anotherfile.txt

sudo adduser username

Q3] Create a group and assign file access using `chgrp`

sudo groupadd groupname

touch groupfile.txt

sudo chgrp groupname groupfile.txt

ls -l groupfile.txt

Q4] Open a file in `vi`, insert text, save, and quit

vi myfile.txt

press i to go in insert mode write someting in it

press ESC then :wq to exit

Q5] In `vi`, use find and replace to change a word in the file

vi myfile.txt

:%s/oldword/newword/g

:wq

Q6] Write a shell script to print your name, roll number, and todayâ€™s date.

First make directory to store all sh files

touch filename.sh

chmod 777

./q6.sh

!/bin/bash

# Define your name and roll number

name="Pranav Prabhat"

roll\_number="1234 5678"

# Get today's date

today\_date=$(date +"%A, %d %B %Y")

# Print the details

echo "Name: $name"

echo "Roll Number: $roll\_number"

echo "Date: $today\_date"

Q7] Write a shell script that accepts two numbers and prints their sum, difference, and product.

#!/bin/bash

read -p "Enter first marks: " a

read -p "Enter second marks: " b

totl=$(($a+$b))

diff=$(($a-$b))

prod=$(($a\*$b))

echo "Total: $totl"

echo "Diffrence: $diff"

echo "Product: $prod"

Q8] Write a script to check whether a file exists and is readable.

/bin/bash

# Prompt user for filename

read -p "Enter the filename to check: " filename

# Check if file exists and is readable

if [ -e "$filename" ] && [ -r "$filename" ]; then

echo "The file '$filename' exists and is readable."

elif [ -e "$filename" ]; then

echo "The file '$filename' exists but is not readable."

else

echo "The file '$filename' does not exist."

fi

Q9] Write a script with a `for` loop to print numbers 1 to 15.

!/bin/bash

# Loop from 1 to 15

for ((i=1; i<=15; i++))

do

echo "$i"

done

Q10] Write a script using `case` to check if a given character is a vowel or consonant.

!/bin/bash

read -p "Enter the alphabet: " alpha

alpha=$(echo "$alpha" | tr 'A-Z' 'a-z')

case "$alpha" in

a|e|i|o|u)

echo "Alphabet is a vowel"

;;

[b-df-hj-np-tv-z])

echo "Alphabet is a consonant"

;;

\*)

echo "Invalid input. Please enter a single alphabet character."

;;

esac

--------------------------------------------------------------------------------------------------------

Assignment 3

Q1] Write a script to calculate factorial of a number using `while` loop.

/bin/bash

read -p "Enter a positive integer: " num

# Validate input

if ! [[ "$num" =~ ^[0-9]+$ ]] || [ "$num" -lt 0 ]; then

echo "Invalid input. Please enter a non-negative integer."

exit 1

fi

factorial=1

counter=$num

while [ $counter -gt 1 ]; do

factorial=$((factorial \* counter))

counter=$((counter - 1))

done

echo "Factorial of $num is: $factorial"

Q2] Write a script to display first 20 even numbers using `for` loop.

/bin/bash

echo "First 20 even numbers are:"

count=0

num=2

while [ $count -lt 20 ]; do

echo "$num"

num=$((num + 2))

count=$((count + 1))

done

Q3] Write a script to print the multiplication table of a number given by the user.

!/bin/bash

# Prompt the user for a number

read -p "Enter a number: " num

echo "Multiplication table for $num:"

# Loop from 1 to 10

for ((i=1; i<=10; i++)); do

result=$((num \* i))

echo "$num x $i = $result"

done

Q4] Write a script to count lines, words, and characters in a file.

!/bin/bash

# Prompt user to enter the filename

read -p "Enter the filename: " filename

# Check if the file exists

if [[ -f "$filename" ]]; then

# Use wc (word count) command to get the counts

lines=$(wc -l < "$filename")

words=$(wc -w < "$filename")

chars=$(wc -m < "$filename")

echo "File: $filename"

echo "Lines: $lines"

echo "Words: $words"

echo "Characters: $chars"

else

echo "Error: File '$filename' not found."

fi

Q5] Write a script to display a menu (date, calendar, list files, exit) and execute the chosen option.

!/bin/bash

while true; do

echo "=============================="

echo " MAIN MENU"

echo "=============================="

echo "1. Display Date"

echo "2. Display Calendar"

echo "3. List Files in Current Directory"

echo "4. Exit"

echo "=============================="

read -p "Enter your choice [1-4]: " choice

case $choice in

1)

echo "Current Date and Time:"

date

;;

2)

echo "Calendar:"

cal

;;

3)

echo "Files in $(pwd):"

ls -l

;;

4)

echo "Exiting... Have a great day!"

break

--------------------------------------------------------------------------------------------------------

ASSIGNMENT 4

ASS 4

1. Write a shell script that accepts a filename as an argument and displays whether it is a directory, a regular file, or something else.

/bin/bash

read -p "Enter your file:" filename

if [ -d "$filename" ]; then

echo "$filename is directory"

elif [ -f "$filename" ];then

echo "$filename is a normmal file"

else

echo " $filename is something else"

fi

2. Write a script that reads a number and prints whether it is positive, negative, or zero.

!/bin/bash

read -p "Enter any number:" num1

if [ "$num1" -gt 0 ]; then

echo "The number is positive"

elif [ "$num1" -lt 0 ]; then

echo "The number is negative"

else

echo " The number is neither positive nor negative"

fi

3. Write a script that accepts a username and checks whether the user is currently logged in.

b!/bin/bash

read -p "Enter user name:" username

if who | grep -w "$username" > /dev/null; then

echo " '$username' is currently logged in."

else

echo "'$username' is not logged in"

fi

4. Write a script that prints the first 10 natural numbers using `until` loop.

!/bin/bash

# Initialize counter

num=1

# Loop until num is greater than 10

until [ "$num" -gt 10 ]

do

echo "$num"

num=$(($num + 1))

done

5. Write a script that demonstrates use of functions by creating one for addition and one for subtraction.

!/bin/bash

add() {

echo "Result of addition: $(( $num1 + $num2 ))"

}

subtract() {

echo "Result of subtraction: $(( $num1 - $num2 ))"

}

read -p "Enter first number: " num1

read -p "Enter second number: " num2

add "$num1" "$num2"

subtract "$num1" "$num2"

--------------------------------------------------------------------------------------------------------

ASSIGNMENT 5

Create git repo

git clone

cd git repo name

nano hello.py

gid add .

git commit

git checkout -b first-branch

git checkout -b second branch

git switch first-branch

nano hello.py

git add.

git commit

git switch second branch

nano hello.py

git add.

git commit

git merge first-branch

nano hello.py

git add .

git commit

git merge first-branch

git switch main

git merge second-branch

git push

sudo apt update

sudo apt install git

git --version

git config -global user.name "prabhat"

git config -global user.email ".com"

git config -- list

mkdir

cd dir

git init

echo "hello"<by.txt

git add .

git commit

git status

git remote add origin githubrepo

git push -u origin master

---------------------------------------------------------------------------------------------------

Assignment 6

Q] Static web hosting

S3 -> BUCKET -> AC1 ENABLED -> ADD INDEX.HTML -> ACTION -> MAKE PUBLIC ACL -> CLICK HTML FILE -> OBJECT URL