Bash sample project

program 1 - In this program I have written code that how to print program on terminal using bash script

#!/bin/bash #this line define the path from which program got executed

echo -e " This is my fisrt program in bash for printing line and comment out line which is not required in code\n" #in this line printing the character over terminal

# using this pound/she/hash is for commenting a single line to tell the interpreter donot execute this . # this is how single line comment is provided on bash script

echo " As we are commeting (in case of single line is by using hash sign) line for us to understand that how things are working and i am using \n but need to assign -e after echo to change the line"

: ' this is the way to comment in multiple lines in the bash code

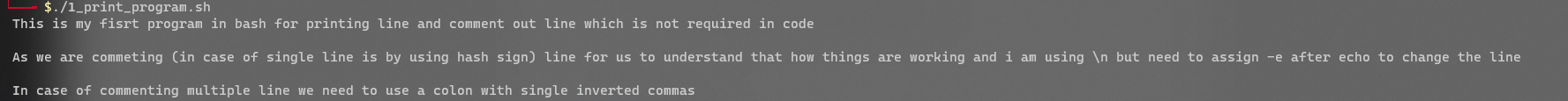
hey you are not a human

i am your best-friend

this is my name bulls' #this I how multi line comment is provided on bash

echo -e "\n In case of commenting multiple line we need to use a colon with single inverted commas"

Output



Program 2 In this program I have define variable in using bash

#!/bin/bash

echo " In this lesson I'll create a variables "

num=20 # way to define variable

echo $num

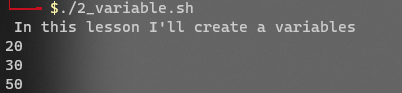
Num=30

echo $Num

add=`expr $num + $Num`

echo $add

Output



Program 3 In this program I have take input from user using bash

#!/bin/bash

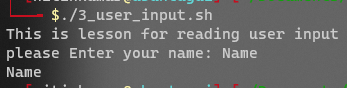
echo -e "This is lesson for reading user input"

echo -e "please Enter your name: \c"

read word

echo $word

Output



Program 4 In this program array is created

#!/bin/bash

echo 'This is the progrm of ARRAY'

ARRAY=('ArchLinux', 'Ubuntu', 'Redhat', OpenSuse) #This is how array is design

ELEMENTS=${#ARRAY[@]} #array defined here

echo ${ARRAY[@]}#print array here

Output



Program 5 decision control (if /else) check weather directory exist or not

#!/bin/bash

echo -e " Program for decision making i.e, if / else statement\n"

directory="./lesson" #directory stored ad var

if [ -d $directory ]; then #if statement is start –d indicates var store directory location

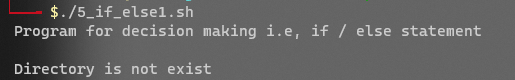
echo " Directory is here"

else

echo " Directory is not exist"

Fi # if end statement

Output



Program 6 decision control (if /else) cheack number is equal or not

#!/bin/bash

echo -e " Program to check the number with if and else statement\n"

num1=39 #dedine number

num2=39

if (($num1 == $num2)); #then comparison of number

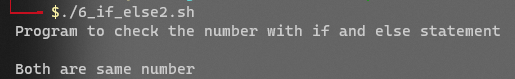
echo " Both are same number "

else

echo " number are not equal"

fi

Output



Program 7 for loop printing numbers

#!/bin/bash

echo -e " Program to print number 1...10 using for loop\n"

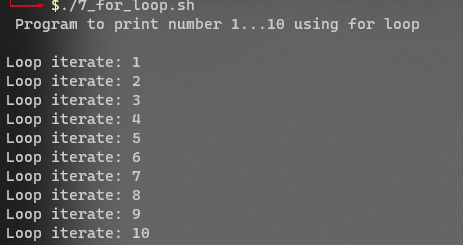
for i in {1..10} #define from where to where

Do #with that print it

echo "Loop iterate:" $i

Done #loop exit

Output



Program 8 while loop doing same as above

#!/bin/bash

echo -e " Program to print number 1...10 using while loop\n"

i=1

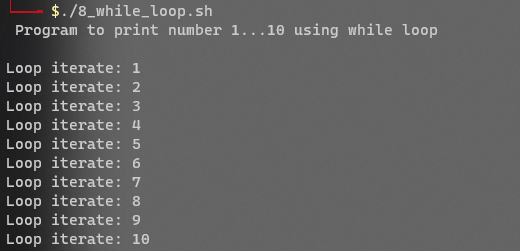
while [ $i -le 10 ];

do

echo "Loop iterate:" $i

((i++))

Done



Program 9 program is working with **grep**

#!/bin/bash

echo -e " Please enter the name of file: \c" #input of file from user

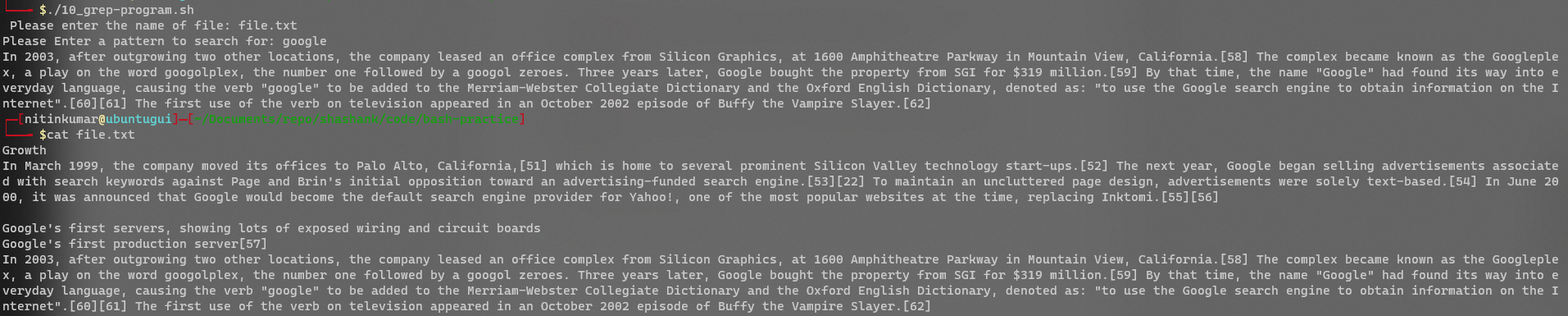
read filename

echo -e "Please Enter a pattern to search for: \c" #pattern form user searching for

read pattern

grep "$pattern" "$filename"

Output :



I enter the file name and the I give pattern (a word) google this give me an output line in which google word is exist.

Program 10 program on **egrep** command

#!/bin/bash

echo -e " Please enter the name of file: \c"

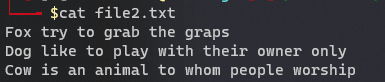
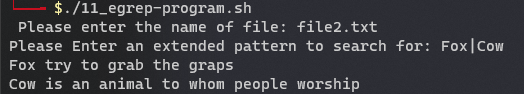
read filename

echo -e "Please Enter an extended pattern to search for: \c"

read ext\_pattern

egrep "$ext\_pattern" "$filename"

Output

  
egrep is also like grep but it is more advanced then grep that it supports a wider range of regular expressions, including the use of alternation, grouping, and quantifiers

In this egrep I have search two different word using a special character that is only supported by egrep

Program 11 awk program

Awk is particularly useful for working with structured data, such as delimited fields and records.

#!/bin/bash

data="

Alice, 25, F

Bob, 32, M

Charlie, 40, M

"

# Print the data to the console

echo "Original data:"

echo "$data"

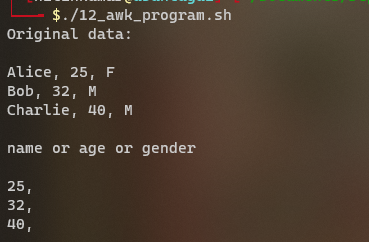
# Use awk to print only the second column of data (the numbers)

echo "name or age or gender"

echo "$data" | awk '{print $2}'

Output :

I want to extract the name and age of all people in the file. We can use the following awk command:



This {print $2} action prints the second fields from data of each line.

Porgram 12 program for lsblk (list of block)

#!/bin/bash

echo -e "Do you want to list all block devices? (y/n) \c"

read answer

if [ "$answer" == "y" ]; then

lsblk

else

echo -e "listing block devices cancelled.\n"

Fi

Output: it just listing the block

