**Write shell script**

1. Accept numbers and perform addition, subtraction, division and multiplication.

gedit arithmetic.sh

echo "Enter value of a:-"

read a

echo "Enter value of b:-"

read b

ansadd=`expr $a + $b`

echo "Addition is " $ansadd

anssub=`expr $a - $b`

echo "Subtraction is " $anssub

ansmult=`expr $a \\* $b`

echo "Multiplication is " $ansmult

ansdiv=`expr $a / $b`

echo "Division is " $ansdiv

1. Accept the string and checks whether the string is palindrome or not.

gedit palindrome.sh

echo "Enter string"

read str

len=`echo $str |wc -c`

i=1

i=$len

while [ $i -gt 0 ]

do

x=`echo $str | cut -c $i`

3 temp=$temp$x

i=`expr $i - 1`

done

echo $temp

if [ $str = $temp ]

then

echo "pallindrom..!!"

else

echo "Not pallindrom..!!"

fi

3. Accept number and check the number is even or odd, finds the length of the number, sum of the digits in the number.

gedit even\_odd.sh

echo " Even no is even or odd"

echo " Eneter a number"

read no

clear

temp=`expr $no % 2`

if [ $temp -eq 0 ]

then

echo $no "is even"

else

echo $no "is Odd"

fi

len=`echo $no | wc -c`

len=`expr $len - 1 `

echo "Length of number is " $len

sum=0

rem=0

while [ $no -gt 0 ]

do

rem=`expr $no % 10`

sum=`expr $sum + $rem`

no=`expr $no / 10`

done

echo "Sum of all digits is:" $sum

1. Accept strings and replace a string by another string.

gedit string\_operation.sh

echo -n "Enter string:"

read str

echo "Yor string is :" $str

echo -n "Enter word u want to replace with other word:"

read oldstr

echo -n "Enter new word:"

read newstr

str=`echo $str | sed s/$oldstr/$newstr/g`

echo "After replacing:" $str

1. Accept filename and displays last modification time if file exists, otherwise display appropriate message.

gedit file\_operation.sh

echo "Enter file name"

read fname

if [ -f $fname ]

then

echo `ls -l|grep $fname|cut -f10 -d ' '`

else

echo "File does not exist"

fi

6. Fetch the data from a file and display data into another file in reverse order.

gedit file\_reverse.sh

echo "fetch data from another file"

ls -a

echo "enetr file name"

read fname

tac $fname > temp

cat temp

1. **Write a script to find the global complete path for any file.**

gedit global\_path.sh

echo "Display Global path for a file" echo "Enter file name"

read fName

echo $fName

echo `pwd`

1. **Write a script to broadcast a message to a specified user or a group of users logged on any**

**terminal.**

gedit broadcast\_message.sh

echo "Enter user Name"

read uName

user=`who | grep -c $uName`

if [ $user -gt 0 ]

then

echo $user

write $uName

else

echo "\nBrodcast message to all loggin users"

echo "\nEnter your message"

echo "\n[Enter ctrl + d to stop]"

wall

fi

1. **Write a script to copy the file system from two directories**

**to a new directory in such a way that only the latest file is copied in case there are common files in both the directories.**

read -p "enter dir name:" dir1

read -p "enter dir2 name:" dir2

if [ ! $dir1 ] || [ ! $dir2 ];then

echo "Directory not exist"

exit 1

fi

ls $dir1 > dir1.txt

ls $dir2 > dir2.txt

mkdir result\_dir

for name1 in `cat dir1.txt`

do

flag=0

for name2 in `cat dir2.txt`

do

if [ "$name1" = "$name2" ];then

flag=1

record=`ls -lt $dir1/$name1 $dir2/$name2 | head -n 1`

cpyfilename=`echo $record|cut -d " " -f9`

`cp $cpyfilename result\_dir`

fi

done

if [ $flag -eq 0 ];then

`cp $dir1/$name1 result\_dir`

fi

done

dir1.txt

rm dir2.txt

1. **Write a script to compare identically named files in two different directories and if they are same, copy one of them in a third directory.**

read -p "enter dir name:" dir1

read -p "enter dir2 name:" dir2

if [ ! $dir1 ] || [ ! $dir2 ];then

echo "Directory not exist"

exit 1

fi

ls $dir1 > dir1.txt

ls $dir2 > dir2.txt

mkdir result\_dir

for name1 in `cat dir1.txt`

do

flag=0

for name2 in `cat dir2.txt`

do

if [ "$name1" = "$name2" ];then

flag=1

record=`ls -lt $dir1/$name1 $dir2/$name2 | head -n 1`

cpyfilename=`echo $record|cut -d " " -f9`

`cp $cpyfilename result\_dir`

fi

done

if [ $flag -eq 0 ];then

`cp $dir1/$name1 result\_dir`

fi

done

rm dir1.txt

rm dir2.txt

1. **Write a script to delete zero sized files from a given directory (and all its sub-directories).**

gedit delete\_zero\_size.sh

echo "DELETE ZERO SIZED FILES"

echo `find . -type f -size 0 -exec rm {} \; `

ls –l

1. **Write a script to display the name of those files (in the given directory) which are having multiple links.**

gedit multiple\_link\_files.sh

echo "Dispaly files with multile links"

echo `find . ! -links 1`

1. **Write a script to display the name of all executable files in the given directory.**

echo "All Executabel Files from current Directory"

find . -perm /u=x,g=x,o=x

1. **Write a script to display the date, time and a welcome message (like Good Morning should be displayed with “a.m.” or “p.m.” and not in 24 hours notation.**

gedit welcome\_message\_time.sh echo "Morning wish"

date

hour=`date|cut -c19-21`

if [ $hour -ge 0 -a $hour -le 12 ]

then

echo "Good Morning..!!"

elif [ $hour -ge 12 -a $hour -le 18 ]

then

echo "Good AfterNoon..!!"

elif [ $hour -ge 18 -a $hour -le 20 ]

then

echo "Good Evening..!!"

else

echo "Good Night..!!"

fi

1. **Write a script to display the directory in the descending order of the size of each file.**

gedit descending\_directory\_files.sh

echo "List of all files in descending order from a directory" echo “Enter Directory “

read direc

cd $direc

ls –Sl

1. **Write a script to implement the following command:**

**Tree (of DOS)**

Gedit tree\_DOS.sh

echo “Tree Command”

tree

1. **Write a script to make following file and directory management operations menu based:**

**Display current directory**

**List directory**

**Make directory**

**Change directory**

**Copy a file**

**Rename a file**

**Delete a file and Edit gedit** file\_&\_directory\_operations.sh echo " 1:Display current Directory"

echo " 2:List Directory"

echo " 3:Make directory "

echo " 4:Change directory "

echo " 5:Copy a file "

echo " 6:Rename a fIle "

echo " 7:Delete a file "

echo " 8:Edit a file "

echo " Enter Your Choice: "

read ch

case $ch in

1)

pwd ;;

2)

ls ;;

3) echo "Enter directory name to make "

read dir\_name

mkdir $dir\_name ;;

4)echo " enter directory to change"

read dir1

`cd $dir1` ;;

5)

echo "Enter two file name to copy"

read f\_name1 f\_name2

cp $f\_name1 $f\_name2

;;

6) echo "Enter old file name to rename"

read fname

echo "Enter new file name to rename"

read nfname

mv $fname $nfname

;;

7)

echo "Enter file name to delete"

read fname

rm $fname ;;

8)echo "Enter file name to edit"

read fname

gedit $fname ;;

\*)echo "You hv entered wrong choice "

;;

esac

**14. Write a script which reads a text file and output the following:**

**Count of character,words and lines.**

**File in reverse. Frequency of particular word in the file.**

**Lower case letter in place of upper case letter.**

gedit file\_operations.sh echo "\n Enter Filename : " read filen

echo "--------------------------------------------------------------" echo " MENU " echo "--------------------------------------------------------------" echo " a) Count the no. of char.s, words, lines."

echo " b) Display a file in reverse."

echo " c) Count the no. of occurrences of a particular word." echo " d) Convert Lower case to UppeCase"

echo "--------------------------------------------------------------"

echo "\n Enter Your Choice : "

read c

case "$c" in

1. echo "\n Total lines,words,char.s "

wc $filen

;;

1. rev $filen > temp // tac file2

cat temp ;;

1. echo "\nEnter word to find :" read w

for i in `cat $filen`

do

echo $i >> f1.txt

done

echo "Total no. of words= \c" ; grep -c $w f1.txt

grep $w f1.txt

rm f1.txt

;;

d)

echo "See $filen file for output..." tr [a-z] [A-Z] < $filen

;;

\*) echo "Invalid choice"

esac

1. **Write a shell script to check whether the named user is currently logged in or not.**

gedit logged\_in\_details.sh

echo "enter user name"

read un

who | grep "^$un" if [ $? -eq 0 ]

then

echo "currently log in"

else

echo "not log in"

fi

1. **Write a script to check whether a given number is palindrome or not.**

gedit number\_palindrome.sh

echo "enter no:-"

read no

temp=$no

rem=0

rev=0

while [ $no -gt 0 ]

do

rem=`expr $no % 10`

rev=`expr $rev \\* 10 + $rem`

no=`expr $no / 10`

done

echo "\n Original no:" $temp

echo "\n reverse no :" $rev

if [ $temp -eq $rev ]

then

echo "No is palindrome"

else

echo "No is not palindrome"

fi

**17. Write a script to display all words of a file in ascending order.**

gedit words\_ascending.sh echo -n "Enter File Name:" read file

if [ -f $file ]

then

for i in `cat $file`

do

`echo $i >> t`

done

else

echo "File Doesn’t Exist..."

fi

sort t

rm t

**18. Write a script to display the last modified file.**

gedit last\_modified\_file.sh echo "Last modified file"

filename=`ls -lt|head -2|tail -1|cut -f10 -d " "`

echo "Last Modified file is:$filename"

**19. Write a shell script to add the statement #include <stdio.h> at the beginning of every C source file in current directory containing printf and fprintf.**

gedit C\_simple.sh

echo "Incluede line #include<stdio.h> in all c file"

cFile=`find . -name "\*.c"`

str="#include<stdio.h>"

for i in $cFile

do

pCount=`grep -c '^printf' $i`

fCount=`grep -c '^fprintf' $i `

iCount=`grep -c '^#' $i `

if [ $pCount -gt 0 -a $fCount -gt 0 -a $iCount -eq 0 ]

then

echo $str > tempc

cat $i >> tempc

rm $i

cp tempc $i

fi

done

**20. Write a script that behaves both in interactive and non-interactive mode. When no arguments are supplied, it picks up each C program from current directory and lists the first 10 lines. It then prompts for deletion of the file. If the user supplies arguments with the script, then it works on those files only.** gedit interactive\_noninteractive.sh

echo "Perform operation as per passed argument"

totalArg=`echo $#`

if [ $totalArg -eq 0 ]

then

echo "No argument supplied"

cFiles=`find . -name "\*.c"`

for i in $cFiles

do

echo "File"$i

echo "First 10 lines of .c file"

head -10 $i

echo "Do you want to remove this file[Yes:Y|No:N]"

read ch

if [ "$ch" = "Y" -o "$ch" = "y" ]

then

echo "User want to delete this file" $i

rm -i $i

else

echo "User Does not want to delete file" $i

fi

done

else

if [ -f $1 ]

then

echo "File Exist"

vi $1

else

echo "File does not Exist"

fi

fi

**21. Write a script that deletes all leading and trailing spaces in all lines in a file. Also remove blank lines from a file. Locate lines containing only printf but not fprintf.**

gedit trailing\_spaces.sh

echo "Remove all leading and trailing space along with blank lines from a given file”

echo "Enter file name"

read fName

if [ -f $fName ]

then

# delete BOTH leading and trailing whitespace from each ine sed 's/^[ \t]\*//;s/[ \t]\*$//' $fName >temp

sed '/^$/d' temp

mv temp $fName

else

echo "File does not exist"

fi