1. Write a shell program to check that whether the file/dir entered by user is exist or not. If it is file then display the contents of the file along with some header and footer text. If it is dir then display the entire contents of that dir only. Else specify if it’s empty.

Ans:

**Input:**

echo enter filename

read file

if [ -f $file ];then

echo yes file exist

cat $file

elif [ -d $file ];then

echo $file is a directory

if [ "$(ls -A $file)" ]; then

echo "directory is not empty"

ls -r -1

else

echo "directory is empty"

fi

else

echo file does not exist

fi

**Output:**

cs17@cs17-desktop:$ chmod u+x prac1

cs17@cs17-desktop:$ ./prac1

enter filename

customeryes file exist

100 | RASHMITHA | ANDHERI

200 | MANITA| MALAD

300 | BHAVIKA| ANDHERI

400 | MANSI |ANDHERI

500 | JAMIL| ANDHERI

3.Write a shell script to accept a number & print its binary equivalent.

Ans:

**Input:**

Read -p " enter the decimal number" b

bin=0

while [ $b -ne 0 ]

do

r=` expr $b % 2 | bc `

b=` expr $b / 2 | bc `

bin=$r$bin

done

bin=`expr $bin/10|bc`

echo $bin

**Output:**

cs17@cs17-desktop:$ chmod u+x prac5.txt

cs17@cs17-desktop:$ ./prac3.txt

enter the decimal number

5

101

4.Write a program to demonstration of command line arguments(positional parameters ).

* 1. **$1, $2** The positional parameters
  2. **$\***  Complete set of positional parameters as a single string
  3. **$#** Number of arguments specified in comand line
  4. **$0** Name of executed command
  5. **$?** Exit status of last command
  6. **$!** PID of last background job
  7. **$@** Same as $\* except when enclosed in double quotes

Ans:

**Input:**

echo Script Name: "$0"

echo Total Number of Argument Passed: "$#"

echo Arguments List -

echo 1. $1

echo 2. $2

echo 3. $3

echo All Arguments are: "$\*"

for i in "$@"

do

echo Argument: $i

done

**Output:**

cs17@cs17-desktop:$ chmod u+x prac4.txt

cs17@cs17-desktop:$ ./prac4.txt 1 2 3

Script Name: ./prac4.txt

Total Number of Argument Passed: 3

Arguments List -

1. 1

2. 2

3. 3

All Arguments are: 1 2 3

Argument: 1

Argument: 2

Argument: 3

5.Write a shell program to find GCD of two numbers

Ans:

**Input:**

echo enter two numbers

read n1 n2

rem=1

t1=$n1

t2=$n2

if [ $n2 -eq 0 ]

then

echo "GCD of $n1 and $n2 = $n1"

exit 0

fi

while [ $rem -ne 0 ]

do

rem=` expr $n1 % $n2 `

n1=$n2

n2=$rem

done

echo "GCD of $t1 , $t2 is $n1"

**Output:**

cs17@cs17-desktop:$ chmod u+x prac5.txt

cs17@cs17-desktop:$ ./prac5.txt

enter two numbers

8 12

GCD of 8 , 12 is 4

6.Write a shell script to create a file which stores the name of files and against each name put either "Morning", "Evening" or "Afternoon" depending upon time when file is created.

Ans:

**Input:**

rm hrs

rm myfile

ls -l|cut -c36-37|tail -n+2>hrs

ls >myfile

for var in `cat hrs`

do

if [ $var -ge 0 -a $var -lt 12 ];then

echo "MORNING" >>msg

elif [ $var -ge 12 -a $var -lt 18 ];then

echo "AFTERNOON" >>msg

elif [ $var -ge 18 -a $var -lt 1e 24 ];then

echo "EVENING" >>msg

else

echo "no data" >>msg

fi

done

paste myfile msg

**Output:**

cs17@cs17-desktop:$ chmod u+x prac6

cs17@cs17-desktop:$ ./prac6

deepak35 MORNING

merit MORNING

myfile MORNING

prac6 AFTERNOON

prac4 MORNING

stud35 EVENING

7.Write a shell program to accept file name and pattern from user. Verify that the pattern is present into the file or not. Display the respective messages and insert some header and footer text.

Ans:-

**Input:**

read -p "enter a file name " file

read -p "enter a pattern " pattern

echo "searching started...."

if grep -e $pattern $file ; then

echo found

else

echo not found

fi

echo "searching complete..."

**Output:**

cs17@cs17-desktop:$ chmod u+x prac7

cs17@cs17-desktop:$ ./prac7

enter a file nameprac1

enter a patternecho

searching started....

echo enter filename

echo yes file exist

echo $file is a directory

echo "directory is not empty"

echo "directory is empty"

echo file does not exist

found

searching complete...

9.Write a shell Program to accept three latter month form user and specify the number of days in to it. For Feb specify if it is leap year.

Ans:

**Input:**

read -p "enter month" month

case $month in

jan|JAN|Jan) echo 31;;

feb|FEB|Feb) read -p "enter year" year

if (( ($year%4==0) && ($year%100!=0) || ($year%400==0) )) ;then

echo 29 days

else

echo 28 days

fi;;

mar|MAR|Mar) echo 31 days;;

apr|APR|Apr) echo 30 days;;

may|MAY|May) echo 31 days;;

jun|JUN|Jun) echo 30 days;;

jul|JUL|Jul) echo 31 days;;

aug|AUG|Aug) echo 31 days;;

sept|SEPT|Sept) echo 30 days;;

oct|OCT|Oct) echo 31 days;;

nov|NOV|Nov) echo 30 days;;

dec|DEC|Dec) echo 31 days;;

esac

**Output:**

cs17@cs17-desktop:$ chmod u+x prac9

cs17@cs17-desktop:$ ./prac9

1.enter month feb

enter year2016

29

cs17@cs17-desktop:$ ./prac9

enter month may

31

10.Write a shell program to print prime number series.

Ans:

**Input:**

read -p "Enter a number: " num

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

do

a=0

for((j=2;j<i;j++))

do

if [ $(( $i % $j )) -eq 0 ] ;then

a=1

fi

done

if [ $a -eq 0 ] ;then

echo $i

fi

done

**Output:**

cs17@cs17-desktop:$ chmod u+x prac10.txt

cs17@cs17-desktop:$ ./prac10.txt

Enter a number: 10

1

2

3

5

7

11.Create a two file one with basic information of product and another for Customers. Write a shell program to allow a user to work with two files customers and product as per the choice entered. Ask the user to enter pro\_id or cus\_id as per their choice of files and in response display their entire details.

Ans: **Input:**

read-p “enter your choice” ch

echo "1.product file "

echo "2.customer file "

case $ch in

1) read -p "enter pid" pid

if grep -e $pid product

then

echo found

else

echo not found

fi;;

2)read -p "enter cid" cid

if grep -e $cid customer

then

echo found

else

echo not found

fi;;

esac

**Output:**

cs17@cs17-desktop:$ chmod u+x prac11

cs17@cs17-desktop:$ ./prac11

1.enter your choice

1.product file

2.customer file

1

enter pid001

001 | laptop | 25000

found

2.enter your choice

1.product file

2.customer file

2

enter cid100

100 | RASH | ANDHERI

found

12. Write a shell script to repeat a given string N-numbers of time.

Ans:

**Input:**

read -p "enter a string:" a

read -p "enter the upper limit:" b

for i in `seq $b`

do

echo $a

done

**Output:**

cs17@cs17-desktop:$ chmod u+x prac12.txt

cs17@cs17-desktop:$ ./prac12.txt

enter a string: a thousand splendid suns

enter the upper limit:5

a thousand splendid suns

a thousand splendid suns

a thousand splendid suns

a thousand splendid suns

a thousand splendid suns

13.Write a shell program for continuation of any file ( for execution ).

Ans:

**Input:**

ans="yes"

while [ $ans == "yes" ] ;

do

read -p "enter shell prog you want to execute" prog

chmod u+x $prog

sh ./$prog

read -p "do u want to continue..." ans

done

**Output:**

cs17@cs17-desktop:$ chmod u+x prac13

cs17@cs17-desktop:$ ./prac13

enter shell prog you want to executeprac9

enter monthjan

31

do u want to continue...no

14. Write a shell program to print table of given number.( using for…var…in….loop)

Ans:

**Input:**

echo "Enter a Number"

read n

i=0

for i in {1..10}

do

echo " $n x $i = `expr $n \\* $i`"

i=`expr $i + 1`

done

**Output:**

cs17@cs17-desktop:$ chmod u+x prac14

cs17@cs17-desktop:$ ./prac14

Enter a Number

2

2 x 1 = 2

2 x 2 = 4

2 x 3 = 6

2 x 4 = 8

2 x 5 = 10

2 x 6 = 12

2 x 7 = 14

2 x 8 = 16

2 x 9 = 18

2 x 10 = 20

15.Write a shell program to accept upper and lower limit from user. Display the table of all the number coming between the these limit (including limits) using for…var..in loop.

Ans:

**Input:**

echo "Enter a Number"

read n

echo "Enter Range"

read r

i=0

for i in `seq $r`

do

echo " $n x $i = `expr $n \\* $i`"

done

**Output:**

cs17@cs17-desktop:$ chmod u+x prac15

cs17@cs17-desktop:$ ./prac15

Enter a Number

2

Enter Range

5

2 x 0 = 0

2 x 1 = 2

2 x 2 = 4

2 x 3 = 6

2 x 4 = 8

2 x 5 = 10