**PRACTICAL NO. 1**

1. **Program to demonstrate if statement:-**

Echo Enter any number

Read n

If test $n –gt 0

Then

Echo $n is positive

Else

Echo $n is negative

Fi

**OUTPUT:-**

Linux:~ sh if1.sh

Enter any number

2

2 is positive

Linux:~ sh if1.sh

Enter any number

-6

-6 is negative

1. **Program to demonstrate if else statement for the existence of a file:-**

echo ENTER THE FILE NAME

read fname

if [ -f $fname]

then

echo THE FILE EXISTS

else

echo FILE WITH THIS NAME DOES NOT EXIST

fi

**OUTPUT:-**

Linux:~ sh fname.sh

Enter the filename

if1.sh

THE FILE EXISTS

Linux:~ sh fname.sh

Enter the filename

If4.sh

FILE WITH THIS NAME DOES NOT EXIST

1. **Program to demonstrate string comparison using if else statement:-**

echo ENTER THE TWO STRINGS

read str1 str2

if [ $str1 = $str2 ]

then

echo THE STRINGS ARE IDENTICAL

else

echo STRINGS ARE NOT IDENTICAL

fi

**OUTPUT:-**

Linux:~ sh string.sh

Enter the two strings

Rads rads

THE STRINGS ARE IDENTICAL

Linux:~ sh string.sh

Enter the two strings

Rads ra

THE STRINGS ARE NOT IDENTICAL

1. **Program to check whether file has permission or not:-**

echo Enter the file name

read fname

if [ -w $fname ]

then

echo Type text to append

cat >> $fname

else

echo The file has no write permission

fi

**OUTPUT:-**

Linux:~ sh fpermission.sh

If1.sh

Type text to append

1. **Program to find the type of the file:-**

echo Enter the name of the file

read fname

case $fname in

\*.c) echo it’s a c prog file;;

\*.out) echo it is an output file;;

\*) echo not sure;;

esac

**OUTPUT:-**

Linux:~ sh type.sh

Enter the name of the file

abc.c

it’s a c prog file

**PRACTICAL NO. 2**

1. **Program to demonstrate menu driven program:-**

echo enter

echo 1 to see the contents of /etc/passwd

echo 2 to see list of users

echo 3 to see present working directory

echo 4 exit

echo enter your choice

read n

case $n in

1. ls /etc;;
2. ls /home;;
3. pwd;;
4. exit;;

\*) echo enter the choice as 1,2,3or4;;

esac

**OUTPUT:-**

Linux:~ sh menu.sh

enter

1 to see the contents of /etc/passwd

2 to see list of users

3 to see present working directory

4 exit

enter your choice

3

/root

1. **Program to print first n numbers and their sum:while loop:-**

i=1

sum=0

while [ $i –le 10 ]

do

echo $i

let sum=” $sum + $i`

let i = ” $i + 1”

done

echo The sum is= $sum

**OUTPUT:-**

Linux:~ sh sum1.sh

1

2

3

4

5

6

7

8

9

10

The sum is = 55

1. **Program to call another file:-**

echo hello!!

echo date is ‘date’

echo call prog15

bash shprog15

**OUTPUT:-**

Linux:~ sh calla.sh

hello!!

date is ‘date’

call prog15

shprog15

1. **Program to check prime or not:-**

Clear

echo "Enter the number"

read num

i=2

flag=0

while [ $i -lt $num ]

do

k=`expr $num % $i`

if test $k -eq 0

then

flag=1

fi

i=`expr $i + 1`

done

if test $flag -eq 0

then

echo "Number is prime"

else

echo "Number is not prime"

fi

**OUTPUT:-**

Linux:~ sh prime.sh

Enter the number

23

Number is prime

1. **Program to print even number series:-**

no=0

echo "Enter limit"

read limit

echo "Even number series"

while [ $no -le $limit ]

do

echo $no

let no=” $no + 2 “

done

**OUTPUT:-**

Linux:~ sh.limit.sh

Enter limit

4

Even number series

0

2

4

**PRACTICAL NO. 3**

1. **Program to print factorial numbers:-**

fact=1

i=1

echo "Enter the number :"

read n

while [ $n -ge $i ]

do

let fact=” $fact \* $i “

let i= ” $i + 1”

done

echo "Factorial = $fact"

**OUTPUT:-**

Linux:~ sh fact.sh

Enter the number

4

Factorial=24

1. **Program to demonstrate simple for loop:-**

echo "For loop demonstration "

for i in 1 2 3; do echo $i; done

echo "For loop demo "

for i in see spot run

do

echo $i

done

echo "For loop demo (of output LINUX) "

for i in L I N U X

do

echo Give me a $i!

echo $i!

done

**OUTPUT:-**

Linux:~ sh for1.sh

For loop demonstration

1

2

3

Linux:~ sh for1.sh

For loop demonstration

Give me a LINUX!

LINUX

1. **Program to demonstrate for loop demo:-**

echo "For loop demo "

for i in $\*

do

echo I know $i

done

**OUTPUT:-**

Linux:~ sh ford.sh

For loop demo

Linux

I know Linux

1. **Program to display multiplication table:-**

echo "For loop demo "

clear

tab=1

i=1

a=10

echo "Input Number :"

read x

echo

echo "Table of $x"

for i in 1 2 3 4 5 6 7 8 9 10

do

let i= ” $x \* $i ”

echo $i

done

**OUTPUT:-**

Linux:~ sh ford2.sh

Input Number:

4

Table of 4

4

8

12

16

20

24

28

32

36

40

1. **Program to enter the marks in 5 subjects and print their grades:-**

echo Enter the marks in the five subjects

scale=2

read m1 m2 m3 m4 m5

let per= ” $m1 + $m2 + $m3 + $m4 + $m5 “

let per=” $per / 5 “

echo $per

if [ $per –lt 35 ]

then

echo GRADE = FAIL

fi

if [ $per –ge 35 –a $per –lt 45 ]

then

echo GRADE = THIRD

fi

if [ $per –ge 45 –a $per –lt 60 ]

then

echo GRADE = SECOND

fi

if [ $per –ge 60 –a $per –lt 75 ]

then

echo GRADE = FIRST

fi

**OUTPUT:-**

Linux:~ sh marks.sh

Enter the marks in the five subjects

55 66 55 66 55

59

GRADE = SECOND

Linux:~ sh marks.sh

Enter the marks in the five subjects

12 25 14 12 32

19

GRADE = FAIL

**PRACTICAL NO. 4**

1. **Program to demonstrate day of week using case ….in:-**

echo Enter the day number

read num

case $num in

1. echo Sunday;;
2. echo Monday;;
3. echo Tuesday;;
4. echo Wednesday;;
5. echo Thursday;;
6. echo Friday;;
7. echo Saturday;;

\*) echo Enter the number bet 1 to 7;;

esac

**OUTPUT:-**

Linux:~ sh case1.sh

Enter the day number

1

Sunday

1. **Program to find the type of the character entered:-**

echo Enter any character

read c

case $c in

[a-z]) echo Small Case letter;;

[A-Z]) echo Capital letter;;

[0-9]) echo digit;;

?) echo special symbol;;

\*) echo more than one character, reenter;;

esac

**OUTPUT:-**

Linux:~ sh char.sh

Enter any character

a

Small case letter

Linux:~ sh char.sh

Enter any character

A

Capital letter

1. **Program to print the summation of n numbers:-**

Clear

echo Enter number

read n

i=1

sum=0

while [ $i -le $n ]

do

let sum= “ $sum + $i ”

let i=” $i + 1 “

done

echo "Summation = $sum"

**OUTPUT:-**

Linux:~ sh sum.sh

Enter number

5

Summation =15

1. **Program to print table:-**

echo "Print table"

j=1

echo Input number

read x

for i in 1 2 3 4

do

let j=” $x \* $i “

echo $j

let i=” $i + 1”

done

**OUTPUT:-**

Linux:~ sh table.sh

Print table

Input number

4

4

8

12

16

1. **Program to demonstrate for loop:-**

echo "For loop demonstration "

echo " "

for i in fly spinder frog

do

echo I know an old lady who swallowed a $i.

echo Swallowed a $i\?

echo Swallowed a $i!

done

**OUTPUT:**

Linux:~ sh fly.sh

For loop demonstration

I know an old lady who swallowed a fly.

Swallowed a fly?

Swallowed a fly!

I know an old lady who swallowed a spinder.

Swallowed a spinder?

Swallowed a spinder!

I know an old lady who swallowed a frog.

Swallowed a frog?

Swallowed a frog!

**PRACTICAL NO. 5**

**1. Write a shell script to accept a number & print its binary equivalent:-**

echo Enter the number

read no

a=$no

s=0

x=1

while(test $a -ne 0)

do

temp=`expr $a % 2`

a=`expr $a / 2`

y=`expr $temp "\*" $x`

s=`expr $s + $y`

x=`expr $x "\*" 10`

done

echo Binary equivalent of decimal number $no is $s

**OUTPUT:-**

Linux:~ sh binary.sh

Enter the number

4

Binary equivalent of decimal number 4 is 100

1. **Program to search a pattern:-**

echo "Enter the pattern to search :"

read pat

echo "Enter the filename :"

read fname

echo OUTPUT

echo -----------------

grep "$pat" $fname && echo "Pattern found in file"

echo "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"

grep "$pat" $fname || echo "Pattern not found"

exit

**OUTPUT:**

Linux:~ sh pattern.sh

Enter the pattern to search

frog

Enter the filename

fly.sh

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

For i in fly spinder frog

Pattern found in fly

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

For I in fly spinder frog

1. **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.**

rm k3

ls -l > k1

cut -c48-50 k1 > k2

for i in `cat k2`

do

if(test $i -gt 0 -a $i -lt 12)

then

echo Morning >> k3

elif(test $i -gt 12 -a $i -lt 18)

then

echo Afternoon >> k3

else

echo Evening >> k3

fi

done

ls > k4

paste k3 k4 > k5

cat k5

**OUTPUT:-**

Linux:~ sh time.sh

Evening 11023

Evening evenodd.sh

Evening file1.sh

Evening frog.sh

Afternoon prime.sh

Afternoon fname 24.sh

Afternoon fname25.sh

Morning fname 26.sh

Morning fname 28.sh

Morning fname 30.sh

Morning fname 16.sh

Morning fname 12.sh

Morning fname 6.sh

1. **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.**

rm mae

ls -l > lnglst

cut -c50-52 lnglst > timelst

for i in `cat timelst`

do

if(test $i -gt 0 -a $i -lt 12)

then

echo Morning >> mae

elif(test $i -gt 12 -a $i -lt 18)

then

echo Afternoon >> mae

else

echo Evening >> mae

fi

done

ls > fnamelst

paste fnamelst mae > mae\_out

cat mae\_out

**OUTPUT:-**

Linux:~ sh time2.sh

11023 Evening

evenodd.sh Evening

file1.sh Evening

frog.sh Evening

prime.sh Afternoon

fname 24.sh Afternoon

fname25.sh Afternoon

fname 26.sh Morning

fname 28.sh Morning

fname 30.sh Morning

fname 16.sh Morning

fname 12.sh Morning

fname 6.sh Morning

1. **Program to print n^y using while loop:-**

echo "Enter the number : "

read n

echo "Enter power : "

read y

res=1

while [ $y -ge 1 ]

do

res=`expr $res \\* $n`

y=`expr $y - 1`

done

echo "Output : $res"

**OUTPUT:-**

Linux:~ sh power.sh

Enter the number

4

Enter the power

2

Output : 16

**PRACTICAL NO. 6**

1. **Program to demonstrate switch case for the month:-**

month=(0 31 29 31 30 31 30 31 31 30 31 30 31)

echo Enter month

read mon

case "$mon" in

[jJ][aA][nN]\*) echo "No.of days : ${month[1]} ";;

[fF][eE][bB]\*) echo enter year

read year

let rem="$year % 4"

if [ $rem -eq 0 ] ; then

echo "No. of days : ${month[2]}"

else

echo "No. of days : 28 "

fi;;

[mM][aA][rR]\*) echo "No. of days : ${month[3]} "

esac

**OUTPUT:-**

Linux:~ sh month.sh

Enter month

Jan

No.of days : 31

Linux:~ sh month.sh

Enter month

Feb

enter year

2004

No.of days : 29

1. **Program to print prime number series.**

Clear

echo Enter limit

read limit

j=1

flag=1

echo "Prime Number Series"

while [ $j -le $limit ]

do

i=2

while [ $i -lt $j ]

do

temp=`expr $j % $i`

if [ $temp -eq 0 ]

then

flag=0

break;

else

flag=1

fi

i=`expr $i + 1`

done

if test $flag -eq 1

then

echo $j

fi

j=`expr $j + 1`

done

**OUTPUT:-**

Linux:~ sh pri.sh

Enter limit

7

Prime Number Series

1

2

3

5

7

1. **Program for continuation of any file ( for execution )**

echo "Do U want to execute file ? (Y/N) "

read ch

while [ $ch = 'y' ]

do

echo Enter filename to execute :-

read filename

cls

cat $filename

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

echo " "

sh $filename

sleep

echo "Do u want to continue ? (Y/N) "

read ch

done

**OUTPUT:-**

Linux:~ sh cont.sh

Do U want to execute file ? (Y/N)

y

Enter filename to execute :-

pri.sh

Enter limit

7

Prime Number Series

1

2

3

5

7

Do u want to continue ? (Y/N)

n

1. **Enter a filename and check:-**

Clear

echo Enter filename :

read file

if [ -s $file ]

then

if [ -f $file ]

then

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

cat $file

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

elif [ -d $file ]

then

ls -i $file

fi

else

echo "This file is empty"

fi

**OUTPUT:-**

Linux:~ sh file.sh

Enter filename :

fly.sh

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

echo "For loop demonstration "

echo " "

for i in fly spinder frog

do

echo I know an old lady who swallowed a $i.

echo Swallowed a $i\?

echo Swallowed a $i!

done

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

1. **Program to print first n numbers and their sum:do…until**

i=1

sum=0

until [ $i –gt 10 ]

do

echo $i

sum=`expr $sum + $i`

i=`expr $i + 1`

done

echo The sum is= $sum

**OUTPUT:-**

Linux:~ dountil.sh

1

2

3

4

5

6

7

8

9

10

The sum is =55

**PRACTICAL NO. 7**

1. **Program to identify alphabets, spaces, name entered or name not entered:**

while echo -n "Name : "

do

read name

case "$name" in

\*[^a-zA-Z]\*)

echo "Can contain only alphabets and spaces"

continue;;

\*)

echo "Name not entered"

continue;;

\*)

echo "$name">>stu

break;;

esac

done

**OUTPUT:-**

Linux:~ sh alpha.sh

Name: 6589

Can contain only alphabets and spaces

Name :

Name not entered

Name : ABC

Linux:~ cat stu

ABC

1. **Program to find the pattern:-**

echo Enter the word

read str

case $str in

[aeiou]\*) echo The word begins with a vowel;;

[0-9]\*) echo The word starts with a digit;;

\*[0-9]) echo The word ends with a digit;;

????) echo The word entered is 4 lettered word;;

\*) echo The word entered is either starts with a

consonant or incorrect input;;

esac

**OUTPUT:-**

Linux:~ sh vowel.sh

Enter the word

event

The word begins with a vowel

Linux:~ sh vowel.sh

Enter the word

1pri

The word ends with a digit

Linux:~ sh vowel.sh

Enter the word

Abcd

The word entered is 4 lettered word

1. **Program to demonstrate for loop for lower and upper limit and print their tables :**

echo "For loop demo "

clear

tab=1

i=1

echo "Input lower limit :"

read x1

echo "Input upper limit :"

read x2

echo j=$x1

while [ $j -le $x2 ]

do

echo "table of $j"

for i in 1 2 3 4 5 6 7 8 9 10

do

i=`expr $j \\* $i`

echo $i

done

j=`expr $j + 1`

done

**OUTPUT:-**

Linux:~ sh upplow.sh

For loop demo

Input lower limit:

4

Input upper limit:

6

table of 4

4

8

12

16

20

24

28

32

36

40

table of 5

5

10

15

20

25

30

35

40

45

50

table of 6

6

12

18

24

30

36

42

48

54

60

1. **Program to demonstrate for loop to print the sum of 1-10 numbers:-**

sum=0

for i in 1 2 3 4 5 6 7 8 9 10

do

sum=`expr $sum + $i`

echo $i

i=`expr $i + 1`

done

echo The sum is= $sum

**OUTPUT:-**

Linux:~ sh forsum.sh

1

2

3

4

5

6

7

8

9

10

The sum is = 55

1. **Program to display the directories and files:-**

echo The directories are

echo

for file in \*

do

if [ -d $file ]

then

echo $file

fi

done

echo The files are

echo

for file in \*

do

if [ -f $file ]

then

echo $file

fi

done

**OUTPUT:-**

Linux:~ sh dir.sh

The directories are

Desktop

Bin

The files are

11023

12023

add.sh

fname.sh

pri.sh

prime.sh

demo.sh

factorial.sh

for1.sh

for3.sh

**PRACTICAL NO. 8**

1. **Program to demonstrate function:-**

function one()

{

echo "Hello How r u?"

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

}

function two()

{

echo "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"

}

echo "This is function call demo"

one

two

echo "Function call over"

**OUTPUT:-**

Linux:~ sh funct.sh

This is function call demo

Hello How r u?

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

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Function call over

1. **Program to check given number is perfect or not.**

Clear

i=1

sum=0

echo Enter number

read n

while [ $i -lt $n ]

do

k=`expr $n % $i`

if test $k -eq 0

then

sum=`expr $sum + $i`

fi

i=`expr $i + 1`

done

if [ $n -eq $sum ]

then

echo The given number is perfect number

else

echo The given number is not perfect

fi

**OUTPUT:-**

Linux:~ sh perfect.sh

Enter number

0

The given number is perfect number

Linux:~ sh perfect.sh

Enter number

1

The given number is not perfect

1. **Program to check number is even or odd.**

echo "Enter the number"

read no

k=`expr $no % 2`

if test $k -eq 0

then

echo "Number is even"

else

echo "Number is odd"

fi

**OUTPUT:-**

Linux:~ sh evenodd.sh

Enter the number

6

Number is even

Linux:~ sh evenodd.sh

Enter the number

5

Number is odd

1. **Program to find gcd of a number:-**

Clear

echo "enter first no :"

read num

echo "enter second no :"

read num1

rem=`expr $num % $num1`

while test $rem -gt 0

do

num1=`expr $num`

num=`expr $num1`

done

echo "gcd="$num1

**OUTPUT:-**

Linux:~ sh gcd.sh

enter first no :

4

enter second no :

2

gcd=2

1. **Program to demonstrate menu items:-**

ans=Y

while [ $ans = "Y" ]

do

echo " MENU"

echo "1 . Query on product"

echo "2 . Query on customer"

echo "0 . Exit"

echo "enter ur choice : "

read ch

case "$ch" in

1. echo "Enter product number :"

read pno

grep $pno masterdata

echo "Query on other product : "

read ans

if(test $ans = "Y")

then

continue

fi;;

1. echo "Enter customer number :"

read cno

grep $cno transdata;;

1. exit;;

1)

\*) echo "Invalid choice. Try again..."

esac

done

**OUTPUT:-**

Linux:~ sh option.sh

MENU

1 . Query on product

2 . Query on customer

0 . Exit

enter ur choice :

0

Linux:~

1. **Program to illustrate date:-**

IFS="-"

n="[0-9][0-9]"

y="[0-9][0-9][0-9][0-9]"

month=(0 31 29 31 30 31 30 31 31 30 31 30 31)

echo Enter date

read val

case "$val" in

" ") echo "No value entered"

;;

$n-$n-$y) set $val

let rem=" $3 % 4 "

if [ $2 -gt 12 -o $2 -lt 1 ]

then

echo "illegal month"

exit

else

case "$val" in

29-02-????) if [ $rem -gt 0 ] ; then

echo "$3 is not a leap year"

echo "Illegal date"

exit

fi;;

\*) if [ $1 -gt ${month[$2]} -o $1 -eq 0 ];then

echo "illegal day"

exit

fi;;

esac

fi;;

\*) echo "Invalid date" ;;

esac

echo "$1-$2-$3 is valid date"

**OUTPUT:-**

Linux:~ sh date.sh

Enter date

31-07-1990

31-07-1990 is valid date

Linux:~ sh date.sh

Enter date

32-07-1990

illegal day

Linux:~ sh date.sh

Enter date

31-13-1990

illegal month