## Ex.NO:1 Date: 15.11.18

# PRIME OR NOT

## Aim:

To write a shell script to check whether a given number is prime or not.

## **ALGORITHM:**

STEP 1: Open terminal and type gedit filename.sh command

- STEP 2: Declare the variable and fetch value from the user using read command
- STEP 3: Use do while loop for checking given number is prime are not
- STEP 4: Print the statement using echo keyword
- **STEP 5:** Give full access permission to that file
- STEP 6: In terminal window type ./filename.sh command for display the output
- **STEP 7:** Stop the process.

## **PROGRAM:**

```
echo "Enter a number: "
read num
i=2
f=0
while [ $i -le `expr $num / 2` ]
do
if [ `expr $num % $i` -eq 0 ]
then
f=1
fi
i=`expr $i+1`
done
if [ $f -eq 1 ]
then
echo "The number is composite"
else
echo "The number is Prime"
fi
```

#### **OUTPUT:**

Enter a number: 5 The number is Prime

## **RESULT:**

## Ex. No: 2 Date: 15.11.18

## **DISPLAY CALENDARS**

## Aim:

To write a shell script to modify call command to display calendars of the specified months.

### **ALGORITHM:**

STEP 1: Open terminal and type gedit filename.sh command

- **STEP 2:** Declare the variable and fetch value from the user using read command
- **STEP 3:** Use if statement to check condition
- STEP 4: Print the statement using echo keyword
- **STEP 5:** Give full access permission to that file
- **STEP 6:** In terminal window type ./filename.sh command for display the output

**STEP 7:** Stop the process.

## **Program:**

```
set `date`
y=$3
if [ $y -le 9 ]
then
cal |sed "s/$3/*/"
else
cal |sed "s/$3/**/"
fi
```

#### **Output:**

oem@linux ~ \$ ./prg2.sh March 2019 Su Mo Tu We Th Fr Sa 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31

### **RESULT:**

# **Ex.NO:3** DISPLAY CALENDARS OF THE SPECIFIED RANGE OF MONTHS Date: 26.11.18

### Aim:

To write a shell script to modify —call command to display calendars of the specified range of months

#### **ALGORITHM:**

STEP 1: Open terminal and type gedit filename.sh command

- STEP 2: Declare the variable and fetch value from the user using read command
- **STEP 3:** Use if statement to print specified months
- STEP 4: Print the statement using echo keyword
- **STEP 5:** Give full access permission to that file
- STEP 6: In terminal window type ./filename.sh command for display the output

**STEP 7:** Stop the process.

#### **Program:**

#!/bin/bash currentmonth=\$(date +"%B") month = (date + %m)leap=\$(date +%Y) M1=\$01 M2 = \$02M3=\$03 M4=\$04 M5=\$05 M6=\$06 M7=\$07 M8=\$08 M9=\$09 M10=\$10 M11=\$11 M12=\$12 M13=\$13 echo "The current month is \$currentmonth" if [ \$month -eq \$M1 ]; then echo "\$month has 31 days" elif [ \$month -eq \$M2 ]; then echo "\$month has 28 days" elif [ \$month -eq \$M3 ]; then echo "\$month has 31 days" elif [ \$month -eq \$M4 ]; then echo "\$month has 30 days" elif [ \$month -eq \$M5 ]; then echo "\$month has 31 days" elif [ \$month -eq \$M6 ]; then echo "\$month has 30 days"

elif [ \$month -eq \$M7 ]; then echo "\$month has 31 days" elif [ \$month -eq \$M8 ]; then echo "\$month has 31 days" elif [ \$month -eq \$M9 ]; then echo "\$month has 30 days" elif [ \$month -eq \$M10 ]; then echo "\$month has 31 days" elif [ \$month -eq \$M11 ]; then echo "\$month has 30 days" elif [ \$month -eq \$M12 ]; then echo "\$month has 31 days" else [ \$[\$leap % 400] -eq "0" ]; then echo "This is a leap year. February has 29 days"

## **OUTPUT:**

The current month is March

## **RESULT:**

# Ex.NO:4 DISPLAY VALID LOGIN NAME Date: 26.11.18

## Aim:

To write a shell script to aCSUept a login name. If not a valid login name display message entered login name is invalid.

## **ALGORITHM:**

STEP 1: Open terminal and type gedit filename.sh command

- STEP 2: Declare the variable and fetch value from the user using read command
- **STEP 3:** Use if statement to show message
- **STEP 4:** Print the statement using echo keyword
- **STEP 5:** Give full access permission to that file
- **STEP 6:** In terminal window type ./filename.sh command for display the output

**STEP 7:** Stop the process.

## **Program:**

#!/bin/sh
name="admin"
echo "Enter Your Role"
read name1
if [ \$name = \$name1 ]
then
echo "You are admin"
else
echo "You are not admin"
fi

#### **OUTPUT:**

Enter Your Role admin You are admin

## **RESULT:**

## Ex.NO: 5 Date: 04.12.18

## **DISPLAY DATE**

### Aim:

To Write a shell script to display date in the mm/dd/yy format..

## **ALGORITHM:**

- STEP 1: Open terminal and type gedit filename.sh command
- STEP 2: Use predefined command to fetch current date
- STEP 3: Print the statement using echo keyword
- **STEP 4:** Give full access permission to that file
- STEP 5: In terminal window type ./filename.sh command for display the output
- **STEP 6:** Stop the process.

## **Program:**

#!/bin/bash CURRENTDATE=`date +"%Y-%m-%d %T"` CURRENTDATEONLY=`date +"%b %d, %Y"` CURRENTEPOCTIME=`date +"%Y-%m-%d %T"` echo Current Date is: \${CURRENTDATEONLY} echo Current Date and Time is: `date +"%Y-%m-%d %T"` echo Current Date and Time is: \${CURRENTDATE} echo Current Unix epoch time is: \${CURRENTEPOCTIME}

## **OUTPUT:**

Current Date is: Mar 06, 2019 Current Date and Time is: 2019-03-06 17:08:23 Current Date and Time is: 2019-03-06 17:08:23 Current Unix epoch time is: 2019-03-06 17:08:23

## **RESULT:**

# Ex.NO:6DISPLAY THE TOTAL NUMBER OF USERSDate: 04.12.18

### Aim:

To write a shell script to display on the screen sorted output of —who command along with the total number of users

## **ALGORITHM:**

- STEP 1: Open terminal and type gedit filename.sh command
- **STEP 2:** Use who command
- STEP 3: Print the statement using echo keyword
- STEP 4: Give full access permission to that file
- STEP 5: In terminal window type ./filename.sh command for display the output
- **STEP 6:** Stop the process.

## **Program:**

#!/bin/bash

#+Your username

- # + The time and date
- #+Who is logged on

# + also output a line of asterices (\*\*\*\*\*\*) after each section

# function to display a line of asterices

function line(){

}

echo "Your username :\$USER"; line # call function

echo "Current date and time :\$(date)";line

echo "Currently logged on users:"

who

line

## **OUTPUT:**

Your username :oem

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

Current date and time :Sat Mar 16 12:38:41 IST 2019

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

Currently logged on users:

oem tty7 2019-03-16 11:53 (:0)

oem pts/0 2019-03-16 12:38 (:0)

## **RESULT:**

## Ex.NO:7 Date: 12.12.18

## **MULTIPLICATION TABLE**

#### Aim:

To write a shell script to display the multiplication table any number.

#### **ALGORITHM:**

STEP 1: Open terminal and type gedit filename.sh command

- STEP 2: Declare the variable and fetch value from the user using read command
- **STEP 3:** Use for loop to print particular range
- STEP 4: Print the statement using echo keyword
- **STEP 5:** Give full access permission to that file
- STEP 6: In terminal window type ./filename.sh command for display the output
- **STEP 7:** Stop the process

#### **Program:**

```
clear
echo enter a number
read n
for i in {1..12}
do
echo "$i*$n"= $(($i * $n))
done
```

### **OUTPUT:**

enter a number 2 1\*2=2 2\*2=4 3\*2=6 4\*2=8 5\*2=10 6\*2=12 7\*2=14 8\*2=16 9\*2=18 10\*2=20 11\*2=2212\*2=24

### **RESULT:**

# **Ex.NO:8** FILE NAME AS INPUT AND DISPLAY WHETHER IT EXITS OR NOT Date: 12.12.18

## Aim:

To write a shell script to accept a file name as input and display whether it exits or not.

### **ALGORITHM:**

- STEP 1: Open terminal and type gedit filename.sh command
- STEP 2: Declare the variable and fetch value from the user using read command
- STEP 3: Use if statement to check file is exit or not
- STEP 4: Print the statement using echo keyword
- **STEP 5:** Give full access permission to that file
- STEP 6: In terminal window type ./filename.sh command for display the output
- **STEP 7:** Stop the process

## **Program:**

clear echo "Enter the filename" read name echo echo "File details" if [-f \$name] then echo "File exists" echo -n "file size is:" wc -c \$name echo -n "File type:" file \$name if [ -r \$name -a -w \$name -a -x \$name ] then echo "File is readable writeable and executable" elif [ -r \$name -a -w \$name ] then echo "File is readable and writable" elif [ -r \$name ] then echo "File is read only" elif [ -w \$name ]

then echo "File is write only " elif [ -x \$name ] then echo "File executable only" else echo "File not readable ,writable and executable" fi else echo "File does not exist" fi

## **OUTPUT:**

Enter the filename file1

File details File exists file size is:27 file1 File type:file1: ASCII text File is readable and writable

## **RESULT:**

# Ex.NO:9SUM OF THE FIRST N NUMBERSDate: 20.12.18

#### Aim:

To write a shell script to find the sum of the first n numbers.

#### **ALGORITHM:**

- STEP 1: Open terminal and type gedit filename.sh command
- STEP 2: Declare the variable and fetch value from the user using read command
- **STEP 3:** Use do while loop to check file is exit or not
- STEP 4: Print the statement using echo keyword
- **STEP 5:** Give full access permission to that file
- STEP 6: In terminal window type ./filename.sh command for display the output

**STEP 7:** Stop the process

#### **Program:**

echo "Enter a number: " read num

i=1 sum=0

```
while [ $i -le $num ]
do
sum=`expr $sum + $i`
i=`expr $i + 1`
done
echo "The sum of first $num numbers is: $sum"
```

## **OUTPUT:**

Enter a number: 5 The sum of first 5 numbers is: 15

## **RESULT:**

# Ex.NO:10COMPARING AND SORTING THE FILE.Date: 20.12.18

## Aim:

To write a shell script to merge the contents of three files, sort the contents and then display them page by page

## **ALGORITHM:**

STEP 1: Open terminal and type gedit filename.sh command

- STEP 2: Declare the variable and fetch value from the user using read command
- STEP 3: Use comm and sort command for comparing and sorting

STEP 4: Print the statement using echo keyword

- **STEP 5:** Give full access permission to that file
- STEP 6: In terminal window type ./filename.sh command for display the output

**STEP 7:** Stop the process

## **Program:**

echo "\*\*\*Comparing and sorting the file\*\*\*"

echo "first file name with extension "

read f1 echo "enter second file name with extension "

read f2

```
comm <(sort f1.txt) <(sort f2.txt)</pre>
```

## **OUTPUT:**

first file name with extension f1.txt enter second file name with extension f2.txt ai are

doing

happen

hi

what

## **RESULT:**

## Ex.NO:11 Date: 08.01.19

# LCD OF TWO NUMBERS

## Aim:

To write a shell script to find the LCD (least common divisor) of two numbers.

## **ALGORITHM:**

- STEP 1: Open terminal and type gedit filename.sh command
- STEP 2: Declare the variable and fetch value from the user using read command
- STEP 3: Use if statement to check the condition of given numbers
- STEP 4: Print the statement using echo keyword
- **STEP 5:** Give full access permission to that file
- STEP 6: In terminal window type ./filename.sh command for display the output
- **STEP 7:** Stop the process

## **Program:**

echo "Enter the first number :" read a echo "Enter the second number : " read b

```
if [ $a -gt $b ]
then
num=$a
den=$b
else
num=$b
den=$a
fi
r=`expr $num % $den`
while [$r -ne 0]
do
num=$den
den=$r
r=`expr $num % $den`
done
gcd=$den
lcm=`expr $a \* $b / $gcd`
```

echo " The LCM of \$a and \$b is : \$lcm" echo " The GCD of \$a and \$b is : \$gcd"

# **OUTPUT:**

Enter the first number :

25Enter the second number :15The LCM of 25 and 15 is : 75

The GCD of 25 and 15 is : 5

## **RESULT:**

## Ex.NO:12 Date: 08.01.19

# CALCULATOR

### Aim:

To write a shell script to stimulate a calculator that does the addition, subtraction, multiplication and division between two numbers. The user should be prompted for two operands and operator.

## **ALGORITHM:**

STEP 1: Open terminal and type gedit filename.sh command

- STEP 2: Declare the variable and fetch value from the user using read command
- STEP 3: Use while loop and switch case for perform Arithmetic operation
- STEP 4: Print the statement using echo keyword
- **STEP 5:** Give full access permission to that file
- STEP 6: In terminal window type ./filename.sh command for display the output

**STEP 7:** Stop the process

#### **Program:**

```
ans='Y'
while [ ans = 'Y' - o  ans = 'y' ]
do
echo "Enter operand1: "
read op1
echo "Enter operand2: "
read op2
echo "Enter operation: "
echo "+ for addition"
echo "- for subtraction"
echo "* for multiplication"
echo "/ for division"
echo "Enter your choice: "
read op
case $op in
"*")sum=`expr $op1 \* op2`
  echo "Product is $sum";;
"+")sum=`expr $op1 + op2`
  echo "Sum is $sum";;
"-")sum=`expr $op1 - $op2`
  echo "Difference is $sum";;
"/")sum=`expr $op1 / $op2`
  echo "Quotient is $sum";;
*)echo "Invalid Entry"
esac
echo "Do you wish to continue:(Y/N):"
read ans
```

done

# OUTPUT

Enter operand1: 8 Enter operand2: 4 Enter operation: + for addition - for subtraction \* for multiplication / for division Enter your choice: / Quotient is 2 Do you wish to continue:(Y/N): n

# **RESULT:**

# Ex.NO:13FIND THE POWER OF GIVEN NUMBERDate: 23.01.19

#### Aim:

To write a shell script that takes two number s through keyboard and finds the value odd one number raised to the power of another.

### **ALGORITHM:**

STEP 1: Open terminal and type gedit filename.sh command

STEP 2: Declare the variable and fetch value from the user using read command

**STEP 3:** Use while loop for finding power of given number

STEP 4: Print the statement using echo keyword

**STEP 5:** Give full access permission to that file

STEP 6: In terminal window type ./filename.sh command for display the output

**STEP 7:** Stop the process

#### **Program:**

echo "Enter a number: " read a

echo "Enter Power: " read p

#### i=1

ans=1 while [ \$i -le \$p ] do ans=`expr \$ans \\* \$a` i=`expr \$i + 1` done echo "Answer of \$a^\$p is \$ans"

#### **OUTPUT**

Enter a number: 10 Enter Power: 3 Answer of 10<sup>3</sup> is 1000

## **RESULT:**

## Ex.NO:14 Date: 23.01.19

# **BINOMIAL COEFFICIENT**

## Aim:

To write a shell script to find the binomial coefficient C(n, x).

## **ALGORITHM:**

- STEP 1: Open terminal and type gedit filename.sh command
- STEP 2: Declare the variable and fetch value from the user using read command
- **STEP 3:** Use while loop for finding power of given number
- **STEP 4:** Print the statement using echo keyword
- **STEP 5:** Give full access permission to that file
- STEP 6: In terminal window type ./filename.sh command for display the output
- **STEP 7:** Stop the process

## **Program:**

factorial() {

[[ "\$1" -lt 2 ]] && echo 1 && return

```
echo $(( $1 * $( factorial $(($1 - 1)) ) ))
```

echo \$(factorial 7)

## **OUTPUT:**

5040

## **RESULT:**

# Ex.NO:15FIND THE LARGEST OF THREE NUMBERSDate: 31.01.19

#### Aim:

To find the largest of three numbers and also find the total and the average

#### **ALGORITHM:**

- STEP 1: Open terminal and type gedit filename.sh command
- STEP 2: Declare the variable and fetch value from the user using read command
- **STEP 3:** Use if statement for finding average
- STEP 4: Print the statement using echo keyword
- **STEP 5:** Give full access permission to that file
- STEP 6: In terminal window type ./filename.sh command for display the output
- **STEP 7:** Stop the process

## **Program:**

echo "Enter the 1st Number: " read n1 echo "Enter the 2nd Number: " read n2 echo "Enter the 3rd Number: " read n3 total = `expr \$n1 + \$n2 + \$n3`avg=`expr \$total / 3` if [ \$n1 -gt \$n2 -a \$n1 -gt \$n3 ] then echo "\$n1 is the Largest of three" elif [ \$n2 -gt \$n1 -a \$n2 -gt \$n3 ] then echo "\$n2 is the Largest of three" elif [ \$n3 -gt \$n1 -a \$n3 -gt \$n2 ] then echo "\$n3 is the Largest of three" else echo "All the three numbers are Equal" fi echo "Total: \$total" echo "Average: \$avg"

OUTPUT Enter the 1st Number: 5 Enter the 2nd Number: 6 Enter the 3rd Number: 7 7 is the Largest of three Total: 18 Average: 6

# **RESULT:**

## Ex.NO:16 Date: 08.02.19

# **FIND A FACTORIAL**

## Aim:

To Write a sccript to find a factorial of a given number.

## **ALGORITHM:**

STEP 1: Open terminal and type gedit filename.sh command

- STEP 2: Declare the variable and fetch value from the user using read command
- STEP 3: Use while loop for finding power of given number
- STEP 4: Print the statement using echo keyword
- **STEP 5:** Give full access permission to that file
- STEP 6: In terminal window type ./filename.sh command for display the output
- **STEP 7:** Stop the process

## **Program:**

fact=1 ie=1

echo -e "Enter a number:\c" read a

```
while [ $ie -le $a ]
do
fact=`expr $fact \* $ie`
ie=`expr $ie + 1`
done
echo -e "Multilpication of $a number is $fact."
```

## **OUTPUT**

Enter a number:10 Multilpication of 10 number is 3628800.

## **RESULT:**

## Ex.NO:17 Date: 08.02.19

# **ARMSTRONG NUMBER**

## Aim:

To write a shell script to check whether a given number is an Armstrong number or not...

## **ALGORITHM:**

STEP 1: Open terminal and type gedit filename.sh command

- STEP 2: Declare the variable and fetch value from the user using read command
- STEP 3: Use while loop for finding power of given number
- STEP 4: Print the statement using echo keyword
- **STEP 5:** Give full access permission to that file
- STEP 6: In terminal window type ./filename.sh command for display the output
- **STEP 7:** Stop the process

## **Program:**

echo "Enter a number: " read c

```
x=$c

sum=0

r=0

n=0

while [ $x -gt 0 ]

do

r='expr $x % 10'

n='expr $r \* $r \* $r'

sum='expr $sum + $n'

x='expr $x / 10'

done
```

if [ \$sum -eq \$c ] then echo "It is an Armstrong Number." else echo "It is not an Armstrong Number." fi

## **OUTPUT:**

Enter a number: 153 It is an Armstrong Number.

## **RESULT:**

# Ex.NO:18 SYMBOLIC MODE/ABSOLUTE MODE Date: 18.02.19

## Aim:

To write a shell Script to assign a file permission to the given file using Symbolic Mode/Absolute Mode

### **ALGORITHM:**

- STEP 1: Open terminal and type gedit filename.sh command
- STEP 2: Declare the variable and fetch value from the user using read command
- **STEP 3:** Use chmod for changing permission of file
- STEP 4: Print the statement using echo keyword
- **STEP 5:** Give full access permission to that file
- STEP 6: In terminal window type ./filename.sh command for display the output

**STEP 7:** Stop the process

## **Program:**

clear echo "Enter the file name " read file echo "Enter the file contents and press control + d" cat>\$file echo echo echo "Permission for the file " ls -l \$file echo echo "Changing permission using symbolic mode" chmod u+x \$file chmod g+wx \$file chmod o-r \$file echo "File permission after assigning permission using symbolic mode" ls -l \$file echo echo "changing permission using absolute mode" chmod u=rw \$file chmod g=rw \$file

chmod o=r \$file echo "File permission after assigning permission using absolute mode" ls -l \$file

## **OUTPUT:**

Enter the file name file3 Enter the file contents and press control + d ub comp unix

Permission for the file -rw-r--r-- 1 shaan51 shaan51 13 2012-03-01 05:08 file3

Changing permission using symbolic mode File permission after assigning permission using symbolic mode

## **RESULT:**