```
**********************************************************************
**********************************************************************
Name: Sheth Preksha
Class: MCA III
Rollno: 36
Subject: Operating System - OS

**********************************************************************
**********************************************************************
Q1)

Basic salary of a person is input through the keyboard. His
dearness allowance is 40% of basic salary and house rent is 20%
of
  basic salary. Write a program to calculate the gross pay.
**********************************************************************
**********************************************************************
echo "Enter Salary : "
read salary

da=$(($salary * 40  / 100))
h_r=$(($salary * 20 / 100))
gross_pay=`expr $salary + $da + $h_r`
echo "Gross Pay Salary  = " $gross_pay

**********************************************************************
**********************************************************************
Output:
preksha@DESKTOP-A0UDC7F:~$ sh a1.sh
Enter Salary :
10000
Gross Pay Salary  =  16000
**********************************************************************
**********************************************************************
Q2)

The distance between two cities is input through the keyboard(in
km). Write a program to convert this distance into metres, feet,
   inches and centimeters and display the results.

**********************************************************************
**********************************************************************
echo "Enter Distance between 2 cities :"
read dist
meter=`expr $dist \* 1000`
echo "Distance in meter = "$meter
```

```
feet=`echo "scale = 2;$dist * 3280.84"| bc`
echo "Distance in Feet = "$feet

inch=`echo "scale = 2;$dist *39370.08"| bc`
echo "Distance in Inches = "$inch

cm=`expr $dist \* 100`
echo "Distance in Centimeter = "$cm
```

****
****

Output:

```
preksha@DESKTOP-A0UDC7F:~$ sh a2.sh
Enter Distance between 2 cities :
125
Distance in meter = 125000
Distance in Feet = 410105.00
Distance in Inches = 4921260.00
Distance in Centimeter = 12500
```

****
****

Q3)

The length and breadth of a rectangle and radius of a circle are entered through the keyboard, calculate the perimeter and area of rectangle and area and circumference of the circle.

****
****

```
echo "Enter length for rectangle :"
read length
echo "Enter breadth for rectangle :"
read breadth

area=`echo "scale = 2;$length * $breadth" | bc`
echo "Area of Rectangle = "$area
peri=`echo "scale = 2;$area * 2" | bc`
echo "Perimeter of rectangle = "$peri
echo "Enter radius for circle :"
read radius

area_c=`echo "scale = 2;3.14 * $radius * $radius" | bc`
echo "Area of circle = "$area_c

cirm=`echo "scale = 2; 2 * 3.14 * $radius" | bc`
```

```
echo "Circumference  of the circle = "$cirm
```

****
****

Output:

```
preksha@DESKTOP-A0UDC7F:~$ sh a3.sh
Enter length for rectangle :
2
Enter breadth for rectangle :
3
Area of Rectangle = 6
Perimeter of rectangle = 12
Enter radius for circle :
3
Area of circle = 28.26
Circumference  of the circle = 18.84
```

****
****

Q4)

If a five digit number is entered through the keyboard, calculate the sum of its digits.

****
****

```
echo "Enter Five digit Number : "
read no
num=$no
sum=0
while [ "$no" -gt 0 ]
do
        rem=$(($no % 10))
        sum=$(($sum + $rem))
        no=$(($no / 10))
done
echo "Sum of "$num" of digits = "$sum
```

****
****

Output:

```
preksha@DESKTOP-A0UDC7F:~$ sh a4.sh
Enter Five digit Number :
12345
Sum of 12345 of digits = 15
```

```
**********************************************************************
**********************************************************************
Q5)

The file /etc/passwd contains info about all users. Write a
program which would receive the logname during execution, obtain
   information about it from the file and display the
information on screen in some appropriate format. (Hint : use
cut)
   eg. Logname : UID : GID : Default working directory : Default
working shell

**********************************************************************
**********************************************************************
cut -f 1,3,4,6,7 -d":" /etc/passwd | tail -n 1

**********************************************************************
**********************************************************************
Output:

preksha@DESKTOP-A0UDC7F:~$ sh a5.sh
preksha:1000:1000:/home/preksha:/bin/bash
**********************************************************************
**********************************************************************
Q6)

The script will receive the filename or filename with its full
path, the script should obtain information about this file as
   given by "ls -l" and display it in proper format.
   eg. Filename : File access permissions : Number of links :
Owner of the file : Group to which belongs : Size of file : File
       modification date : File modification time

**********************************************************************
**********************************************************************
echo "Enter File Name : "
read fname

ls -l $fname | cut -d ' ' -f 1,2,3,4,5,6,7,8,9

or

echo "Enter File Name : "
read fname

ls -l $fname | cut -d ' ' -f 1,2,3,4,5,6,7,8,9 |awk '{print $9
":" $1 ":" $2 ":" $3 ":" $4 ":" $5 ":" $6 ":" $7 ":" $8}'
```

```
**********************************************************************
**********************************************************************
Output:
preksha@DESKTOP-A0UDC7F:~$ sh a6.sh
Enter File Name :
a1.sh
-rw-r--r-- 1 preksha preksha 169 Nov 20 22:46 a1.sh


or


preksha@DESKTOP-A0UDC7F:~$ sh a6.sh
Enter File Name :
a1.sh
a1.sh:-rw-r--r--:1:preksha:preksha:169:Nov:20:22:46
**********************************************************************
**********************************************************************
Q7)

If cost price and selling price of an item are entered through
the keyboard, write a program to determine whether the seller
has
   made profit or loss. Also determine how much profit/loss is
made.

**********************************************************************
**********************************************************************
echo "Enter Cost price : "
read cp
echo "Enter Selling price : "
read sp

if [ $sp -gt $cp ]
then
        echo "The Seller Made profit of "`expr $sp - $cp`
else
        echo "The seller made loss of "`expr $cp - $sp`
fi

**********************************************************************
**********************************************************************
Output:

preksha@DESKTOP-A0UDC7F:~$ sh a7.sh
```

```
Enter Cost price :
100
Enter Selling price :
120
The Seller Made profit of 20


preksha@DESKTOP-A0UDC7F:~$ sh a7.sh
Enter Cost price :
120
Enter Selling price :
100
The seller made loss of 20
```

****

****

Q8)

Check whether the entered no. is odd or even.

****

****

```
echo "Enter Number : "
read no

rem=`expr $no % 2`
if [ $rem -eq 0 ]
then
        echo "$no is even number.."
else
        echo "$no is odd number.."
fi
```

****

****

Output:

```
preksha@DESKTOP-A0UDC7F:~$ sh a8.sh
Enter Number :
12
12 is even number..

preksha@DESKTOP-A0UDC7F:~$ sh a8.sh
Enter Number :
11
11 is odd number..
```

****

****

Q9)

Check whether the entered no. is prime or not.

```
**************************************************************
**************************************************************
echo "Enter Number : "
read no

flag=1
i=2

while [ $i -lt $no ]
do
        rem=`expr $no % $i`
        #echo "$rem is..."
        if [ $rem -eq  0 ]
        then
                flag=0
                break
        fi
        i=`expr $i + 1`
done

if [ $flag -eq 1 ]
then
        echo "$no is prime Number."
else
        echo "$no is not prime Number."
fi
**************************************************************
**************************************************************
```

Output:

```
preksha@DESKTOP-A0UDC7F:~$ sh a9.sh
Enter Number :
6
6 is not prime Number.


preksha@DESKTOP-A0UDC7F:~$ sh a9.sh
Enter Number :
5
5 is prime Number.
**************************************************************
**************************************************************
```

Q10)

Check whether the entered year is a leap year or not.

```
**************************************************************
**************************************************************
echo "Enter Year : "
read yr

if [ `expr $yr % 4` -eq 0 -a `expr $yr % 100` -ne 0 -o `expr $yr
% 400` -eq 0 ]
then
        echo "$yr is leap year."
else
        echo "$yr is not leap year."
fi

**************************************************************
**************************************************************
```

Output:

```
preksha@DESKTOP-A0UDC7F:~$ sh a10.sh
Enter Year :
2019
2019 is not leap year.


preksha@DESKTOP-A0UDC7F:~$ sh a10.sh
Enter Year :
2020
2020 is leap year.
**************************************************************
**************************************************************
```

Q11)

The script receives two file names as arguments, the script must check whether the files are same or not, if they are similar
    then delete the second file.

```
**************************************************************
**************************************************************
cmp -s $1 $2`
if [ $? -eq 0 ]
then
        echo "$1 and $2 are same files"
        rm $2
        echo "$2 file deleted"
```

```
else
        echo "$1 and $2 are not same files"
fi
```

****
****

Output:

```
preksha@DESKTOP-A0UDC7F:~$ sh a11.sh demo2.txt demotest.txt
demo2.txt and demotest.txt are not same files
```

****
****

Q12)

Write a script which will display whether your friend has logged in or not, if he has logged in then send him some message.

****
****

```
echo "Enter User Name :"
read uname
who | grep $uname > /dev/null

if [ $? -eq 0 ]
then
        echo "User is Logged in.."
        echo "Please enter message :"
        read msg
        echo $msg
else
        echo "User is not logged in .."
fi
```

****
****

Output:

```
[ec2-user@ip-172-31-93-145 ~]$ sh a12.sh
Enter User Name :
preksha
User is Logged in..
Please enter message :
hii
hii
```

```
**************************************************************
**************************************************************
```
Q13)

While executing a shell script, either the logname or uid is supplied at the command prompt, write a shell script to find out at
how many terminals has this user logged in.

```
**************************************************************
**************************************************************
if [ $# -eq 1 ]
then
          total=`who | grep -c $1`
          echo "$1 logged in on total $total terminals"
else
          echo "please enter user name..."
fi
**************************************************************
**************************************************************
```
Output:

```
preksha@DESKTOP-A0UDC7F:~$ sh a13.sh
please enter user name...
preksha@DESKTOP-A0UDC7F:~$ sh a13.sh preksha
preksha logged in on total 1 terminals
**************************************************************
**************************************************************
```
Q14)

Write a shell script to display the date with the format :- 25th October 2005 is a Tuesday.

```
**************************************************************
**************************************************************
date +"%dth %B %Y is a %A"
**************************************************************
**************************************************************
```
Output:

```
preksha@DESKTOP-A0UDC7F:~$ sh a14.sh
08th December 2020 is a Tuesday

**************************************************************
**************************************************************
```

Q15)

Write a shell script to display the appropriate message like :
Good Morning / Good Afternoon / Good Evening

```
**************************************************************
**************************************************************
time=`date '+%H'`
echo "$time"

if [ $time -ge 6 -a $time -lt 12 ]
then
        echo "Good Morning.."
elif [ $time -ge 12 -a $time -lt 16 ]
then
        echo "Good Afternoon.."
elif [ $time -ge 16 -a $time -lt 20 ]
then
        echo "Good Evening.."
else
        echo "Good Night.."
fi
**************************************************************
**************************************************************
```

Output:

```
preksha@DESKTOP-A0UDC7F:~$ sh a15.sh
16
Good Evening..
**************************************************************
**************************************************************
```

Q16)

Write a shell script to display the menu driven interface :- 1) list all files of the current directory, 2) print the current directory, 3) print the date, 4) print the users otherwise display "Invalid Option".

```
**************************************************************
**************************************************************
while [ 1 ]
do
        echo -e "\n\n1.List of all the current
directory\n2.Print the current directory\n3.print the date
\n4.print the users\n0.exit"

        echo "Enter your  choice :"
```

```
        read ch

        case $ch in
                "1") ls ;;
                "2") pwd ;;
                "3") date ;;
                "4") who ;; # awk -F: '{print $1}' /etc/passwd
                "0") echo "Exit"
                     break ;;
                 *) echo "Invalid choice.."
        esac
done
```

****
****

Output:

```
[ec2-user@ip-172-31-93-145 ~]$ sh a16.sh


1.List of all the current directory
2.Print the current directory
3.print the date
4.print the users
0.exit
Enter your  choice :1
a16.sh  emp.dat  first.sh  mca06  sample.dat  sample.sh
second.sh  stud.dat  third.sh


1.List of all the current directory
2.Print the current directory
3.print the date
4.print the users
0.exit
Enter your  choice :2
/home/ec2-user


1.List of all the current directory
2.Print the current directory
3.print the date
4.print the users
0.exit
Enter your  choice :3
Sun Nov 22 08:12:42 UTC 2020
```

```
1.List of all the current directory
2.Print the current directory
3.print the date
4.print the users
0.exit
Enter your  choice :4
ec2-user pts/0        2020-11-22 07:55 (43.241.144.141)


1.List of all the current directory
2.Print the current directory
3.print the date
4.print the users
0.exit
Enter your  choice :5
Invalid choice..


1.List of all the current directory
2.Print the current directory
3.print the date
4.print the users
0.exit
Enter your  choice :0
Exit
```

****
****

Q17)

Create a menu driven calculator which asks for two integers and perform basic arithmetic operations.

****
****

```
echo "Enter !st Integer :"
read no1
echo "Enter 2nd Integer :"
read no2

while [ 1 ]
do
        echo
"\n1.Addition\n2.Subtraction\n3.Multiplication\n4.Division\n0.Ex
it\n"
        echo "Enter Your Choice :"
```

```
        read ch

        case $ch in
                "1") ans=`expr $no1 + $no2`
                        echo "Addition = " $ans ;;
                "2") ans=`expr $no1 - $no2`
                        echo "Subtraction = " $ans ;;
                "3") ans=`expr $no1 \* $no2`
                        echo "Multiplication = " $ans ;;
                "4") ans=`expr $no1 \/ $no2`
                        echo "Division = " $ans ;;
                "0") echo "Exit.."
                        break ;;
                *) echo "Invalid Choice.."
        esac
done
```

****

****

Output:

```
preksha@DESKTOP-A0UDC7F:~$ sh a17.sh
Enter !st Integer :
40
Enter 2nd Integer :
20

1.Addition
2.Subtraction
3.Multiplication
4.Division
0.Exit

Enter Your Choice :
1
Addition =  60

1.Addition
2.Subtraction
3.Multiplication
4.Division
0.Exit

Enter Your Choice :
3
Multiplication =  800

1.Addition
```

```
2.Subtraction
3.Multiplication
4.Division
0.Exit

Enter Your Choice :
4
Division =  2

1.Addition
2.Subtraction
3.Multiplication
4.Division
0.Exit

Enter Your Choice :
2
Subtraction =  20

1.Addition
2.Subtraction
3.Multiplication
4.Division
0.Exit

Enter Your Choice :
u
Invalid Choice..

1.Addition
2.Subtraction
3.Multiplication
4.Division
0.Exit

Enter Your Choice :
5
Invalid Choice..

1.Addition
2.Subtraction
3.Multiplication
4.Division
0.Exit

Enter Your Choice :
0
```

```
Exit..
**********************************************************************
**********************************************************************
```
Q18)

Find the factorial of any number.

```
**********************************************************************
**********************************************************************
echo "Enter Number : "
read no

fact=1

while [ $no -gt 1 ]
do
        fact=$(($fact * $no))
        no=$(($no - 1))
done
echo "Factorial = $fact"


or


echo "Enter Number : "
read no
num=$no
fact=1

while [ $no -gt 1 ]
do
        fact=`expr $fact \* $no`
        no=$(($no - 1))
        #echo "$no+1! is $fact.."
done
echo "Factorial = $fact"
~

**********************************************************************
**********************************************************************
Output:

preksha@DESKTOP-A0UDC7F:~$ sh a18.sh
Enter Number :
5
Factorial = 120
```

```
**********************************************************************
**********************************************************************
```
Q19)

Display the fibonacci series upto some number.

```
**********************************************************************
**********************************************************************
echo "Enter number :"
read num

n1=0
n2=1
i=2

echo "Fibonacci Series : "
echo "$n1\n$n2"

while [ $i -lt $num ]
do
        i=`expr $i + 1`
        n3=`expr $n1 + $n2`
        echo $n3
        n1=$n2
        n2=$n3
done

**********************************************************************
**********************************************************************
```
Output:

```
preksha@DESKTOP-A0UDC7F:~$ sh a19.sh
Enter number :
5
Fibonacci Series :
0
1
1
2
3
**********************************************************************
**********************************************************************
```
Q20)

Two numbers are entered through the keyboard, find the power, one number raised to another.

```
**********************************************************************
**********************************************************************
echo "Enter Power : "
read pow
echo "Enter Exponent : "
read exp

i=0
ans=1

while [ $i -lt $pow ]
do
        ans=`expr $ans \* $exp`
        i=`expr $i + 1`
done


echo "$exp ^ $pow = $ans"
~
**********************************************************************
**********************************************************************

Output:

preksha@DESKTOP-A0UDC7F:~$ sh a20.sh
Enter Power :
2
Enter Exponent :
5
5 ^ 2 = 25
**********************************************************************
**********************************************************************

Q22)
Write a script which reports name and size of all files in a
directory. whose
sizes exceed 1000. The filenames should be printed in the
descending order of
their sizes. The total         no. of files must be reported.

**********************************************************************
**********************************************************************

ls --sort=size -l | awk '$5 >= 1000 {print $5,$9}'

**********************************************************************
**********************************************************************
```

Output:

```
preksha@DESKTOP-A0UDC7F:~$ sh a22.sh
4096 mydir
1496 a16.sh
1392 a17.sh
1059 MCA3_36_Preksha.sh
```

****

****

Q24)

Print the prime nos. from 1 to 300.

****

****

```
        if [ $n -le 3 ]
        then
                return 1
        fi

        if [ $(($n%2)) -eq 0 ]
        then
                return 0
        fi

        if [ $(($n%3)) -eq 0 ]
        then
                return 0
        fi
        i=5

        while [ $(($i*$i)) -eq 0 ]
        do
                if [ $(($n%$i)) -eq 0 ]
                then
                        return 0
                fi
                if [ $(($n%($i+2))) -eq 0 ]
                then
                        return 0
                fi
                i=$((i+6))
        done
        return 1
}

num=2
```

```
while [ $num -le 300 ]
do
        checkprime $num
        isprime=$?

        if [ $isprime -eq 1 ]
        then
                echo "$num"
        fi

        num=$(($num+1))
done
```

****
****

Output:

```
preksha@DESKTOP-A0UDC7F:~$ sh a24.sh
2
3
5
7
11
13
17
19
23
25
29
31
35
37
41
43
47
49
53
55
59
61
65
67
71
73
77
```

79
83
85
89
91
95
97
101
103
107
109
113
115
119
121
125
127
131
133
137
139
143
145
149
151
155
157
161
163
167
169
173
175
179
181
185
187
191
193
197
199
203
205
209
211
215
217

```
221
223
227
229
233
235
239
241
245
247
251
253
257
259
263
265
269
271
275
277
281
283
287
289
293
295
299
```

****
****

Q25)

Program must display all the combinations of 1, 2, and 3.

****
****

```
for i in 1 2 3
do
        for j in 1 2 3
        do
                for k in 1 2 3
                do
                        echo "$i $j $k"
                done
        done
done
```

****
****
Output:

```
preksha@DESKTOP-A0UDC7F:~$ sh a25.sh
1 1 1
1 1 2
1 1 3
1 2 1
1 2 2
1 2 3
1 3 1
1 3 2
1 3 3
2 1 1
2 1 2
2 1 3
2 2 1
2 2 2
2 2 3
2 3 1
2 3 2
2 3 3
3 1 1
3 1 2
3 1 3
3 2 1
3 2 2
3 2 3
3 3 1
3 3 2
3 3 3
```

****
****
Q26)
Write a script for renaming each file in the directory such that it will have
the current shell PID as an extension. The shell script should ensure that
the directories do not get renamed.
****
****

```
for f in *
do
    [ -e $f ] || continue
```

```
        mv $f $f.$$
done
```

****

****

Output:

```
preksha@DESKTOP-A0UDC7F:~$ ls -l
total 0
-rwxrwxrwx 1 neel neel 13 Dec  9 16:38 Hello.sh.82
-rwxrwxrwx 1 neel neel 58 Dec  9 16:38 file1.txt.82
-rwxrwxrwx 1 neel neel  0 Dec  9 14:42 neel.sh.82
-rwxrwxrwx 1 neel neel  0 Dec  9 16:43 s_s_26.sh
-rwxrwxrwx 1 neel neel 55 Dec  9 16:39 s_s_26.sh.82
```

****

****

Q27)
A file called wordfile consists of several words. Write a shell script which
 will receive a list of filenames, the first of which would be wordfile.
The shell script should report all occurences of each word in
wordfile in the rest of the files supplied as arguments.

****

****

```
 if [ $# -eq 0 ]; then
        printf "Usage:\n"
        echo "./27-findWordFromFile.sh <wordFile> <findFile ...>"
        exit
 fi

 filesToRead=$(($#-1))
 echo $filesToRead

 # Reading Line by Line
 while read line; do
# Reading Word by Word
        for word in $line; do
                echo "Searching word: '$word' ..."
                # 2 is slice starting index
                # filesToRead is slice length
                grep --color=always -n $word ${@:2:filesToRead}
```

```
                printf "Done.\n\n"
        done
done <"$1" # $1 is the file name we want to search
```

****

****

Output:

```
preksha@DESKTOP-A0UDC7F:~$ sh a27.sh wordFile.txt findFile.txt
1
Searching word: 'samsung' ...
~
```

****

****

Q28)
Write a shell script which deletes all the lines containing the word "unix"
in the files   supplied as arguments to it.

****

****

```
word="UNIX"

# Read all args
for i; do
        # I is for Insensitive
        # d is for delete
        # I must be written first
        sed -i "/\b$word\b/Id" $
done
```

****

****

Output:

```
preksha@DESKTOP-A0UDC7F:~$ cat unix.txt
unix
helloi
i love unix
linux is best

preksha@DESKTOP-A0UDC7F:~$ sh a28.sh unix.txt

Output :

preksha@DESKTOP-A0UDC7F:~$ cat unix.txt
hello
linux is best
```

```
**************************************************************
**************************************************************

Q29)
The word "unix" is present in only some of the files syupplied
as arguments
to the shell  script. You script should search each of  these
files in turn
and stop at the first file that it encounters containing the
word unix.
The filename should be displayed on the screen.
**************************************************************
**************************************************************
for i
do
        echo "Searching file : $i..."

        if grep -q "unix" "$i";then
                echo "Found in $i"
                exit
        fi
        echo "done"
done

**************************************************************
**************************************************************
Output:

preksha@DESKTOP-A0UDC7F:~$ sh a29.sh temp.txt
Searching file : temp.txt...
Found in temp.txt

preksha@DESKTOP-A0UDC7F:~$ cat temp.txt
hello
unix
how r u
unix

**************************************************************
**************************************************************

Q30)
A shell script receives even number of filenames. Suppose four
filenames are
supplied then  the first file should get copied into
second file, the
```

```
third file should get copied into fourth and so on.. If odd
number of
filenames are supplied display error message
********************************************************************
********************************************************************
#zero arguments

if [ $# -eq 0 ]
then
        echo "No arguments"
        exit
fi

prevFile=$1

#if even no of args

if [ $(echo $# % 2 | bc) -eq 0 ]
then
        #looping through each argument
        count=1
        for i
        do
                if !(($count%2))
                then
                        cp $prevFile $i
                        echo "'$prevFile' copied to -> $i"
                else
                        prevFile=$i
                fi
                count=$(echo $count+1 | bc)
        done
#if odd no of args
else
        echo "Odd no of arguments"
        exit
fi

********************************************************************
********************************************************************

Output:

preksha@DESKTOP-A0UDC7F:~$ cat temp.txt
hello
unix
how r u
```

```
unix

preksha@DESKTOP-A0UDC7F:~$ cat newdemo.txt
prekshasheth
fgdffgef
fdgfdvd
preksha@DESKTOP-A0UDC7F:~$ cat demo.txt
preksha/sheth
fgdf/fgef
fdgf/dvd
preksha@DESKTOP-A0UDC7F:~$ cat demotest.txt
preksha/sheth
fgdf/fgef
fdgf/dvd
preksha@DESKTOP-A0UDC7F:~$ cat demo2.txt
```

****
****

Q31)
The script displays a list of all files in the current directory to which
you have read,   write and execute permissions.

****
****

```
echo "The list of File Names in the current Directory"
echo "Which have read,write and execute permission"

for file in *
do
        if [ -f $file ]
        then
               if [ -r $file -a -w $file -a -x $file ]
               then
                       ls $file
                fi
        fi
done
```

****
****

Output:

```
preksha@DESKTOP-A0UDC7F:~$ sh a31.sh
The list of File Names in the current Directory
Which have read,write and execute permission
```

a1.sh

****

****

Q32)

The script receives any number of filenames as arguments. It should check whether every argument supplied is a file or directory. If it is a directory it should be reported. If it is a filename then name of the file as well as the number of lines present in it should be reported.

****

****

```
for i; do
        if [ -d $i ]; then
                echo "$i -> directory"
        elif [ -f $i ]; then
                printf "$i -> file with lines: "
                wc -l $i | awk {'print $1'}
        else
                echo "$i -> Invalid"
        fi
done
```

****

****

Output:

```
preksha@DESKTOP-A0UDC7F:~$ sh a32.sh demo.txt
demo.txt -> file with lines: 3
```

****

****

Q33)

A script will receive any number of filenames as arguments. It should check whether such files already exist. If they do, then it should be reported, if not then check if a subdirectory "mydir" exists or not in the current directory, if it doesnt exist then it should be created and in it the files supplied as arguments should be created.

****

****

```
if [ $# -eq 0 ]; then
         echo "No Arguments passed"
```

```
         exit
fi

for i; do
# If file exists
        if [ -f $i ]; then
                echo "$i exists"

        else
        # if "mkdir" exists
                if [ -d "mydir" ]; then
                # Directory exists
                printf "directory exists.."
                else
                        mkdir mydir
                fi
                touch mydir/$i
                echo "$i file created in \"mydir\""
        fi
done
```

****
****

Output:

```
preksha@DESKTOP-A0UDC7F:~$ sh a33.sh preksha
directory exists..preksha file created in "mydir"
```

****
****

Q34)
Accept the marks of 5 subjects and calculate the percentage and grade.

****
****

```
echo "Enter Marks for subject 1 : "
read s1
echo "Enter Marks for Subject 2 : "
read s2
echo "Enter Marks for Subject 3 : "
read s3
echo "Enter Marks for Subject 4 : "
read s4
echo "Enter Marks for Subject 5 : "
read s5

total=`expr $s1 + $s2 + $s3 + $s4 + $s5`
```

```
echo "Total = $total"

per=$(($total * 100 / 500 ))
echo "Percentage = $per"

if [ $per -gt 80 ]
then
        echo "Grade is A.."
elif [ $per -lt 80 -a $per -gt 60 ]
then
        echo "Grade is B.."
elif [ $per -lt 60 -a $per -gt 40 ]
then
        echo "Grade is c.."
else
        echo "Fail.."
fi
```

****

****

Output:

```
preksha@DESKTOP-A0UDC7F:~$ sh a34.sh
Enter Marks for subject 1 :
80
Enter Marks for Subject 2 :
89
Enter Marks for Subject 3 :
87
Enter Marks for Subject 4 :
78
Enter Marks for Subject 5 :
80
Total = 414
Percentage = 82
Grade is A.
```

****

****

Q35)
Print armstrog nos. from 1 to 500.

****

****

```
i=1
while [ $i -lt 500 ]
do
```

```
        j=$i
        total=0
        while [ $j -gt 0 ]
        do
                temp=$(echo $j%10 | bc)
                sum=$(echo $temp^3 | bc)
                total=$(echo $total+$sum | bc)
                j=$(echo $j/10 | bc)
        done
        if [ $total -eq $i ]
        then
                echo "Armstrong number : " $i
        fi
        i=`expr $i + 1`
done
```

****
****

Output:

```
preksha@DESKTOP-A0UDC7F:~$ sh a35.sh
Armstrong number :  1
Armstrong number :  153
Armstrong number :  370
Armstrong number :  371
Armstrong number :  407
```

****
****

Q36)
Accept the measure (angles) of a triangle and displa the type of triangle.
(eg. acute, right,obtuse)

****
****

```
echo "Enter angle "
read angle

if [ $angle -ge 0 -a $angle -lt 90 ]
then
        echo "Acute angle"
elif [ $angle -eq 90 ]
then
        echo "Right angle "
elif [ $angle -gt 90 -a $angle -le 180 ]
then
```

```
        echo "Obtuse angle "
else
        echo "Incorrect input"
fi
```

****

****

Output:

```
preksha@DESKTOP-A0UDC7F:~$ sh a36.sh
Enter angle
120
Obtuse angle
```

****

****

Q37)

Display all the numbers from 1 to 100 which are divisible by 7.

****

****

```
checkDivisible(){
          n=$1
          if [ $((n % 7)) -eq 0 ]; then
                return 1
          fi
                return 0

}

num=1

while [ $num -le 100 ]
do
        checkDivisible $num
        isDivisible=$?

        if [ $isDivisible -eq 1 ]
        then
                printf "$num "
        fi

                num=$((num+1))
done
```

****

****

Output:

```
preksha@DESKTOP-A0UDC7F:~$ sh a37.sh
7 14 21 28 35 42 49 56 63 70 77 84 91 98
```

```
**********************************************************************
**********************************************************************
Q38)
Find the largest and smallest of 3 different numbers.

**********************************************************************
**********************************************************************
echo "Enter 1st Number :"
read no1
echo "Enter 2nd Number :"
read no2
echo "Enter 3rd Number :"
read no3

if [ $no1 -gt $no2 -a $no1 -gt $no3 ]
then
        echo "$no1 is Largest.."
elif [ $no2 -gt $no1 -a $no2 -gt $no3 ]
then
        echo "$no2 is Largest.."
else
        echo "$no3 is Largest.."
fi


if [ $no1 -lt $no2 -a $no1 -lt $no3 ]
then
        echo "$no1 is Smallest.."
elif [ $no2 -lt $no1 -a $no2 -lt $no3 ]
then
        echo "$no2 is Smallest.."
else
        echo "$no3 is Smallest..."
fi

**********************************************************************
**********************************************************************

Output:

preksha@DESKTOP-A0UDC7F:~$ sh a38.sh
Enter 1st Number :
3
Enter 2nd Number :
1
Enter 3rd Number :
2
```

```
3 is Largest..
1 is Smallest..
**********************************************************************
**********************************************************************

Q39)
Find HCF and LCM of a given no.
**********************************************************************
**********************************************************************
echo -n "Enter first number : "
read num1
echo -n "Enter second number : "
read num2

max=$num1
den=$num2

if [ $num2 -gt $max ]
then
        max=$num2
        den=$num1
fi

        rem=$((max % den))

while [ $rem -ne 0 ]
do
        max=$den
        den=$rem
        rem=$((max % den))
        max=$((max - 1))
done

        gcd=$den
        lcm=`expr $num1 \* $num2 / $gcd`

echo "HCF of $num1 and $num2 = $gcd"
echo "LCM of $num1 and $num2 = $lcm"

**********************************************************************
**********************************************************************
Output:

preksha@DESKTOP-A0UDC7F:~$ sh a39.sh
Enter first number : 15
Enter second number : 55
HCF of 15 and 55 = 5
```

```
LCM of 15 and 55 = 165
**********************************************************************
**********************************************************************
Q40)
Display the dates falling on Sundays of the current month.
**********************************************************************
**********************************************************************
echo "Sundays in current month are:"

echo " ----- Using AWK ----- "
cal | awk 'FNR > 2{print $1}'


#Sundays in current month are:
 #----- Using AWK -----
 #1
 #6
 #13
 #20
 #27

**********************************************************************
**********************************************************************
Output:

Sundays in current month are:
----- Using AWK -----
1
6
13
20
27
**********************************************************************
**********************************************************************
```