

1. Program of Odd Even Checking.....

```
echo -n "Enter numnber : "  
read n  
rem=$(( $n % 2 ))  
if [ $rem -eq 0 ]  
then  
    echo "$n is even number"  
else  
    echo "$n is odd number"  
fi
```

2.Program of if else checking

```
a=10  
b=20  
if [ $a == $b ]  
then  
    echo "a is equal to b"  
else  
    echo "a is not equal to b"  
fi
```

3. Program for checking prime or not prime.

```
number=43  
i=2  
f=0  
while test $i -le `expr $number / 2`  
do  
if test `expr $number % $i` -eq 0  
then  
f=1  
fi  
i=`expr $i + 1`  
done  
if test $f -eq 1  
then  
echo "Not Prime"
```

```
else
echo "Prime"
fi
```

4. Write a shell script to accept a login name. If not a valid login name display message – “Entered login name is invalid”.

```
echo "Enter The Login Name:"
read username
echo "Enter The Password:"
read password

if [[ $username == "user" && $password == "password" ]]
then
  echo -e "You're Logged In\n"
elif [ $username != "user" ]
then
  echo -e "Invalid User Name\n"
else
  echo -e "Invalid Password\n"
fi
```

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

Open a terminal and type the following date command:

```
$ date +"%m-%d-%y"
```

Sample outputs:

```
02-11-19
```

6. Write a shell script to display on the screen sorted output of “who” command along with the total number of users

The basic syntax is as follows:

```
who
who am i
who [options] [File]
who --help
who --version
who | grep 'userNameHere'
```

Where,

- If no non-options provided, who displays the following information for each user currently logged on:
 - login name
 - terminal line
 - login time
 - remote hostname or X display
- If you give one non-option argument, who uses that instead of a default system-maintained file such as `/var/run/utmp` as the name of the file containing the record of users logged on.
- If given two non-option arguments, who prints only the entry for the user running it preceded by the hostname. Traditionally, the two arguments given are `'am i'`, as in `'who am i'`

- **who command options**

Option	Description
-a	Same as <code>-b -d -login -p -r -t -T -u</code>
-b	Time of last system boot
-d	Print dead processes
-H	Print line of column headings
-l	Print system login processes
-m	Only hostname and user associated with stdin
-p	Print active processes spawned by init
-q	All login names and number of users logged on

-r	Print current runlevel
-t	Print last system clock change
-T	Add user's message status as +, - or ?
-u	List users logged in

7. Shell Program to Print Multiplication Table upto Given Range

```

echo "Enter a Number"
read n

echo "Enter Range"
read r
i=0
while [ $i -le $r ]
do
    echo " $n x $i = `expr $n \* $i`"
    i=`expr $i + 1`
done

```

Output:

```

sunny@sunny-desktop~$ ./mul.sh
Enter a Number: 5
Enter Range: 10
x 0 = 0
x 1 = 5
x 2 = 10
x 3 = 15
x 4 = 20
x 5 = 25
x 6 = 30
x 7 = 35
x 8 = 40
x 9 = 45
x 10 = 50
sunny@sunny-desktop~$

```

8. Write a shell script to compare two files and if found equal asks the user to delete the duplicate file.

```

$ cat duplicate_file.sh
echo "Enter file 1:"
read file1

```

```

echo "Enter file 2:"
read file2
cmp $file1 $file2 > newfile
x=` wc newfile | cut -d" " -f2 `
if [ ` $x -eq 0 `]
then
rm -i $file2
fi

```

9. Write a shell script to find the sum of digits of a given number

```

echo -n "Enter number : "
read n

# store single digit
sd=0

# store number of digit
sum=0

# use while loop to caclulate the sum of all digits
while [ $n -gt 0 ]
do
    sd=$(( $n % 10 )) # get Remainder
    n=$(( $n / 10 )) # get next digit
    sum=$(( $sum + $sd )) # calculate sum of digit
done
echo "Sum of all digit is $sum"

```

10. Write a shell script to merge the contents of three files, sort the contents and then display them page by page

```

file1=$1
file2=$2
file3=$3
out="output.$$"
count=0

if [ $# -ne 3 ]
then
echo "$(basename $0) file1 file2 file3"
exit 1
fi

if [ ! -f $file1 ]
then
echo "$file1 not a file!"
exit 2

```

```
fi
```

```
if [ ! -f $file2 ]  
then  
  echo "$file2 not a file!"  
  exit 3  
fi
```

```
if [ ! -f $file3 ]  
then  
  echo "$file3 not a file!"  
  exit 4  
fi
```

```
cat $file1 $file2 $file3 >> $out  
count=$(cat $out | wc -w)  
echo "$count words written to $out"
```

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

Shell program to find LCM and hcf of two no.s

```
tput clear  
echo "Enter first no"  
read a  
echo "Enter 2nd no"  
read b  
p= 'expr $a \* $b'  
while [ $b -ne 0 ]  
do  
  r= 'expr $a % $b'  
  a=$b  
  b=$r  
done  
LCM = 'expr $p / $a'  
echo "LCM = $LCM"  
echo "Hcf = $a"
```

12. Write a shell script to perform the tasks of basic calculator.

```

clear
sum=0
i="y"

echo " Enter one no."
read n1
echo "Enter second no."
read n2
while [ $i = "y" ]
do
echo "1.Addition"
echo "2.Subtraction"
echo "3.Multiplication"
echo "4.Division"
echo "Enter your choice"
read ch
case $ch in
1)sum=`expr $n1 + $n2`
echo "Sum ="$sum;;
2)sum=`expr $n1 - $n2`
echo "Sub = "$sum;;
3)sum=`expr $n1 \* $n2`
echo "Mul = "$sum;;
4)sum=`expr $n1 / $n2`
echo "Div = "$sum;;
*)echo "Invalid choice";;
esac
echo "Do u want to continue ?"
read i
if [ $i != "y" ]
then
exit
fi
done

```

13. Write a shell script to find the power of a given number.

```

echo "Input number"
read no
echo "Input power"
read power

counter=0
ans=1
while [ $power -ne $counter ]

```

```
do
    ans=`expr $ans \* $no`
    counter=`expr $counter + 1`
done

echo "$no power of $power is $ans"
```

```
Input number
3
Input power
2
3 power of 2 is 9
```

14. Write a shell script to find the factorial of a given number.

```
echo "Enter a number"
read num

fact=1

while [ $num -gt 1 ]
do
    fact=$((fact * num))
    num=$((num - 1))
done
echo $fact
```

15. Write a shell script to check whether the number is Armstrong or not

```
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
```

16. Write a shell script to check whether the file have all the permissions or not.

```
echo -n "Enter file name : "
read file

# find out if file has write permission or not
[ -w $file ] && W="Write = yes" || W="Write = No"

# find out if file has excute permission or not
[ -x $file ] && X="Execute = yes" || X="Execute = No"

# find out if file has read permission or not
[ -r $file ] && R="Read = yes" || R="Read = No"

echo "$file permissions"
echo "$W"
echo "$R"
echo "$X"
```

17. Program to show the pyramid of special character “*”.

```
#Bash Shell Script to print half pyramid using *
rows=4
for((i=1; i<=rows; i++))
do
  for((j=1; j<=i; j++))
  do
    echo -n "*"
  done
  echo
done
```