1. Shell script to test whether the given year is a leap year or not

```
if [ $# -gt 1 ]
then
echo "syntax is <$0> [<year>]"
exit 1
fi
if [ $# -ne 1 ]
then
yr=`date +%Y`
else
yr=$1
fi
d4=`expr $yr % 4`
d100=`expr $yr % 100`
d400=`expr $yr % 400`
if [ $d4 -eq 0 -a $d100 -ne 0 -o $d400 -eq 0 ]
then
echo " $yr is a leap year"
else
echo " $yr is not a leap year"
fi
```

2. Shell script to read 3 numbers and test whether they can be sides of a triangle

```
if [ $# -gt 1 ]
then
echo "Syntax error"
exit 1
fi
if [ $# -eq 0 ]
then
echo "Enter 1st no:"
read a
echo "Enter 2nd no:"
read b
echo "Enter 3rd no:"
read c
fi
if [ `expr $a + $b` -le $c -o `expr $a + $c` -le $b -o `expr $b + $c` -le $a ]
#if [ `expr $a + $b` -le $c ]
```

```
then
echo "$a $b $c cannot be sides of triangle"
else
echo "$a $b $c can be sides of triangle"
fi
```

3. Shell script using command line to check triangle is equilateral isosceles or scalene

```
if [ $# -gt 1 ]
then
echo "Syntax error"
exit 1
fi
if [ $# -eq 0 ]
then
echo "Enter 1st no:"
read a
echo "Enter 2nd no:"
read b
echo "Enter 3rd no:"
read c
fi
if [ `expr $a` -eq $b -a `expr $b` -eq $c ]
then
echo "Equilateral Triangle"
elif [ `expr $a` -eq $b -o `expr $b` -eq $c ]
then
echo "Isoscless Triangle"
else
a2=`expr $a \* $a`
b2=`expr $b \* $b`
c2=`expr $c \* $c`
if [ `expr $a2 + $b2` -eq $c2 ]
then
echo "Right angled Triangle"
else
echo "Scalene Triangle"
fi
```

4. Shell script using command line to test whether given number is prime or not

```
if [ $# -ne 2 ]
```

```
then
echo "syntax is $0 <no1> <no2>"
fi

a=$1
b=$2
echo "\n the prime number between $a and $b are:"
if [ $a -gt $b ]
then
t=$a
a=$b
b=$t
fi

while [ $a -le $b ]
do
i=2
while [ $i -le `expr $a / 2` ]
do
if [ `expr $a % $i` -eq 0 ]
then
break
fi
i=`expr $i + 1`
done
if [ $i -gt `expr $a / 2` ]
then
echo -n "\n$a\n"
fi
a=`expr $a + 1`
done
exit 0
```

5. Shell script to rename all ordinary files in a directory with filename.PID of shell

```
iif [ $# -gt 1 ]
then
    echo "Syntax is <$0> <location> or <$0>"
    exit 1
fi
if [ $# -ne 0 ]
then
    cd $1
```

```
fi
for i in *
do
    if [ -f $i ]
    then
        mv $i $i.$$
        echo "file renamed to .$$ "
    fi
done
```

6. Shell script to compare two files. If the files are identical delete one

```
if [ $# -ne 2 ]
then
echo "syntax is <$0> <file1> <file2>"
elif [ -f $1 -a -f $2 ]
then
cmp $1 $2 >/dev/null
if [ $? -eq 0 ]
then
echo "the file $1 and $2 are same"
rm -f $1
echo "$1 deleted"
else
echo "$1 and $2  are not same"
fi
else
echo "files are not ordinary files"
fi
```

7. Shell script to count the number of words and lines in a file

```
if [ $# -gt 1 ]
then
echo "Syntax is <$0> [<filename>]"
exit 1
fi
flag=0
if [ $# -eq 1 ]
then
term=`tty`
exec < $1
flag=1
```

```
fi
while read line
do
no1=`expr $no1 + 1`
set $line>/dev/null
nowds=`expr $nowds + $#`
done
echo "number of lines=$no1"
echo "number of words=$nowds"
if [ $flag -eq 1 ]
then
exec < $term
fi
exit 0
```

8. Shell script to display message “Good morning”,”Good afternoon”,”Good evening” and “Good night” according to system time

```
hour=$(date +"%H")

if [ "$hour" -ge 5 ] && [ "$hour" -lt 12 ]
 then
   echo "Good morning"
elif [ "$hour" -ge 12 ] && [ "$hour" -lt 17 ]
then
   echo "Good afternoon"
elif [ "$hour" -ge 17 ] && [ "$hour" -lt 21 ]
then
   echo "Good evening"
else
   echo "Good night"
fi
```

9. Shell script to display a file from last line to first line

```
if [ $# -ne 1 ]
then
echo "syntax is <$0> <filename>"
exit 1
fi
if [ -d $1 ]
```

```
then
echo "$1 is a dictionary"
exit 2
fi
cp $1 file
l=`cat file | wc -l`
revfname="$1.rev"
rm $revfname 2>/dev/null
while [ $l -ge 1 ]
do
tail -n 1 file >> $revfname
l=`expr $l - 1`
head -n $l file > temp
mv temp file
done
```

10. Shell script to delete all lines if a file containing word linux

```
if [ $# -ne 2 ]
then
echo " syntax is <$0><string><filename>"
exit 1
fi
term=`tty`
exec<$2
while read line
do
echo $line | grep $1>/dev/null
echo $line | grep -w $1>/dev/null
if [ $? -ne 0 ]
then
echo $line >>temp
fi
done
exec<$term
mv temp $2
```

11. Shell script to display the content of a directory file in windows style

```
echo "\n\n Directory of `pwd` \n"

for f in *
do
```

```
#dpart=`ls -l $f | cut -d ' ' -f6,7`
fdatetime=`date -r $f "+%d-%m-%y %H:%M:%S"`

if [ -d $f ]
then
echo "$f \t\t<DIR>\t\t $fdatetime"
else
f_size=`cat $f | wc -c`
echo "$f \t\t $f_size \t $fdatetime"
fi
done
```

12. Shell script to print specified range of lines from given files

```
if [ $# -le 2 ]
then
   echo "syntax is <$0> <start_line> <end_line> <files>"
   exit 1
fi

# Get range
n1=$1
n2=$2
shift
shift

# Calculate number of lines to extract
n3=$(expr $n2 - $n1 + 1)

# Process each file
while [ $# -ne 0 ]
do
   echo -e "\nlines from $n1 to $n2 from $1\n"
   head -n $n2 "$1" | tail -n $n3
   shift
done

exit 0
```

13. Shell script to copy a set of files given as pairs at command prompt

```
#!/bin/bash
# Check if the number of arguments is even
r=`expr $# % 2`
```

```
if [ $r -ne 0 ]
then
   echo "filenames are not in pairs"
   exit 1
fi

# Loop through arguments in pairs
while [ $# -ne 0 ]
do
   # Copy source file to destination
   cp "$1" "$2"
   echo "Copied '$1' to '$2'"
   shift
   shift
done

exit 0
```

14. Shell script to count the occurrence of words in a file in a set of files

```
if [ $# -eq 0 ]
 then
   echo "Syntax: $0 <word> <file1> <file2> ..."
   exit 1
fi

word=$1
shift
count=0
while [ $# -ne 0 ]
do
for wd in `cat $1`
do
if [ $wd = $word ]
then
count=`expr $count + 1`
fi
done
shift
done
echo "no.of occurance of words = $count"
```

15. Shell script to modify cp command considering all error possibilities

```
if [ $# -ne 2 ]
then
echo "\n syntax is <$0> <src filename> <tgt filename>"
exit 1
fi

if [ ! -f $1 ]
then
echo "$1 is not existing or not an ordinary file"
exit 2
fi

if [ -f $2 ]
then
echo "target file exists,overwrite it (y/n)"
read ans
if [ $ans="n" ]
then
exit 3
fi
fi
cp $1 $2
echo "file copied"
exit 0
```

16. Shell script to perform a menu driven program to display
    a) All ordinary files
    b) All directory files
    c) All special files
    d) All files readable to UGO
    e) All files writable to UGO
    f) All files executable to UGO

```
opt=0

while [ $opt -le 7 ]

do

clear
```

```
echo " \n 1.all ordinary files \n2.all directory files \n3.all special files \n4. all files readable to UGO \n5.all files witable to UGO \n6.all files executable to UGO \n7.all files with read write execute to owner \n8.exit"

echo -n "option:"

read opt


case $opt  in


1)echo "\n ordinary files are:"

for file in *

do

if [ -f $file ]

then

echo $file

fi

done

;;


2)echo "\n directory files are:"

for file in *

do

if [ -d $file ]

then

echo $file

fi

done
```

```
;;

3)echo "\n special files are:"
for file in *
do
if [ ! -d $file -a ! -f $file ]
then
echo $file
fi
done
;;

4)echo "\n all files readable to UGO"
for file in *
do
if [ -f $file ]
then
user=`ls -l $file | cut -c 2-2`
group=`ls -l $file | cut -c 5-5`
owner=`ls -l $file | cut -c 8-8`
if [ $user="r" -a $group="r" -a $owner="r" ]
then
echo $file
fi
fi
```

```
done
;;

5)echo "\n files that are writtable  by UGO :\n"
for file in *
do
if [ -f file ]
then
user=`ls -l $file | cut -c 3-3`
group=`ls -l $file | cut -c 6-6`
owner=`ls -l $file | cut -c 9-9`
if [ $user="w" -a $group="w" -a $owner="w" ]
then
echo $file
fi
fi
done
;;

6)echo "files executable by UGO are:"
for file in *
do
if [ -f file ]
then
user=`ls -l $file | cut -c 4-4`
```

```
group=`ls -l $file | cut -c 7-7`
owner=`ls -l $file | cut -c 10-10`
if [ $user="x" -a $group="x" -a $owner="x" ]
then
echo $file
fi
fi
done
;;

7)echo " files rwx by owner are:"
for file in *
do
if [ -f file ]
then
rwx=`ls -l $file | cut -c 2-4`
if [ $rwx="rwx" ]
then
echo $file
fi
fi
done
;;

8)echo " existing..."
```

```
exit 0

;;


esac

echo -n "enter to continue:"

read x

done
```