FILE   EMP.LST

```
101|ajay|sales|manager|25000|12/10/1990
102|asha|marketing|g.m|30000|10/02/2020
103|mr. shah|purchase|cleark|15000|10/05/2001
104|amar|testing|tester|25000|10/10/2002
105|raj|admin|clerk|40000|05/10/2010
106|rahul|sales|manager|25000|12/10/1990
107|mr. patel|marketing|g.m|30000|10/02/2020
208|mr.  shah|purchase|cleark|15000|10/05/2001
109|amar|testing|tester|25000|10/10/2002
210|MR. Agarwal|admin|clerk|40000|05/10/2010
111|Ajay agrawal|testing|tester|25000|10/10/2002
112|Anil aggarwal|purchase|cleark|15000|10/05/2001
```

FILE DEPT

```
sales
account
account
sales
marketing
purchase
purchase
testing
admin
personnel
```

1. **WRITE A SHELL SCRIPT TO FIND PATTERN (SALES FROM EMP.LST FILE)**

```
if test $# -eq 0
then
  echo "Not Enough Arguments"
elif test $# -eq 2
then
   echo "Pattern Found"
   grep $1 $2
fi
```

2. **#w.a.s to take input as a file name and display total no.of lines words and charcters**

```
echo "Enter file name :"
read fname
l=`wc -l $fname`
c=`wc -c $fname`
w=`wc -w $fname`
echo "No. of lines are :$l"
echo "No.of Characters are :$c"
echo "No. of words are:$w"
```

3.

#take input as a file name and display 3rd column in sorted order

```
cut -d"|" -f 3 $1 | sort -u| tee sort1
```

4. **WRITE A SHELL SCRIPT TO CHECK OLDER AND NEWER FILE , MAKE OLD FILE AS NEW**

```
echo "enter old file name:"
read old
echo "enter new file name:"
read new
```

```
echo "file name with date and time"

ls -lt $old $new

touch $old

echo "List of file name after changes"

ls -l $old $new
```

5. **WRITE A SHELL SCRIPT TO FIND PATTERN USING COMMAND LINE ARGUMENT**

```
if test $# -eq 0

then

  echo "Not Enough Arguments"

elif test $# -eq 2

then

     echo "Pattern Found"

         grep $1 $2

fi
```

6. **WRITE A SHELL SCRIPT TOFIND GREATER AND LESS VALUE**

```
echo "enter a value of a"

read a

echo "enter a value of b"

read b

if test $a -eq $b

then

   echo equall

elif test $a -gt $b

then

     echo "$a is greater than $b"

elif test $a -lt $b

then

     echo "$a is less than $b"
```

```
else
        echo "not equall"
fi
```

7.

```
#chk given pattern is match or not
echo "enter file name"
read fname
echo "enter pattern 1"
read p1
if  grep $p1 $fname
then
   echo found
else
   echo not found
fi
```

8.

```
#chk given pattern is match or not
echo "enter file name"
read fname
echo "enter pattern 1"
read p1
echo "enter pattern 2"
read p2
if  grep $p1 $fname
then
   echo "$p1 found"
elif grep $p2 $fname
then
   echo "$p2 found"
else
```

```
    echo "not found"

fi
```

9. **WRITE A SHELL SCRIPT TO check value equall or not**

```
echo "enter a value of a"
read a
echo "enter a value of b"
read b
if test $a -eq $b
then
    echo eqall
else
   echo "not equall"
fi
```

string1

```
# check string comparision operator
if [ -z "$1" -o -z "$2" ]
then
    echo "String is null"
    elif [ $1 = $2 ]
    then
        echo "strings are Euall"
        elif [ $1 != $2 ]
        then
            echo "Strings are not equall"
            else
                echo "string is not null"
fi
```

file1

```
#file test attribute
if [ $# -eq 1 ]
then
if [ ! -e $1 ]
then
   echo "File does not exist"
elif [ -f $1 ]
then
   echo "File exist"
else
   echo "wrong file name"
fi
else
   echo "file name not given"
fi
```

file 2:   check file attribute

```
echo "enter file name"
read fname
if [ -r $fname ]
then
   echo "File is readable"
fi
if [ -w $fname ]
then
   echo "file is writable"
fi
if [ -x $fname ]
```

```
then
   echo "file is executble"
fi
```

file 3

```
#check file is directory or not , and display no. of files available in it
if [ $# -eq 1 ]
then
   if [ -d $1 ]
   then
     echo "$1 is a directory"
     echo "Number of file available in it"
     ls -l $1 | grep "[^d]" | wc -l
   else
      echo "$1 is not a directory"
   fi
else
  echo "not enough arguments"
fi
```

file 4

```
# check file is newer or older
if [ $# -eq 2 ]
then
   if [ $1 -nt $2 ]
   then
     echo "$1 is newer than $2"
     touch $2
     ls -l $1 $2
     echo "newer file is $2"
   else
```

```
    ls - l $1 $2
    echo "$1 is older than $2"
     touch $1
     ls -l $1 $2
     echo "older file is $2"
   fi
else
   echo "not enough arguments"
fi
```

write a shell script for mathematical calculation (expr)

```
echo "enter value1"
read a
echo "enter value2"
read b
sum=`expr $a + $b`
echo "addition is :$sum"
sub=`expr $a - $b`
echo "subtraction is :$sub"
mul=`expr $a \* $b`
echo "Multiplication is :$mul"
div=`expr $a / $b`
echo "division is :$div"
modulus=`expr $a % $b`
echo "Modulus is :$modulus"
```

example of switch case case1

```
#multichioce case
echo "  Menu   "
echo "1.display list of files"
echo "2.User Process"
echo "3.Today's date"
echo "4.no. of users"
echo "5.quit"
echo "enter Your choice:"
read ch
case "$ch" in
    1) ls -l;;
    2) ps -f;;
    3) date ;;
    4) who ;;
    5) exit ;;
    *) echo "invalid option"
Esac
```

example of While

```
#multichioce case
while [ true ]
do
echo "  Menu   "
echo "1.check string is null or not"
echo "2.find lenght"
echo "3.subtring"
echo "4.display basename"
echo "5.quit"
echo "enter Your choice:"
read ch
case "$ch" in
    1) echo "Enter a string"
       read stg
       if [ -z "$stg" ]
       then
          echo "null string"
```

```
      else
        echo "not null"
      fi
      ;;
    2) echo "enter a string"
      read stg
      echo "length of string is `expr "$stg" : '.*'`"
      ;;
    3) echo "enter a string"
      read stg
      echo "sub string is `expr "$stg" : '......\(...\)'`"
      ;;
    4) echo "enter a filename :"
    read fname
    echo "extract basename"
    read chr
       echo "basename is ::`basename $fname $chr`"
       ;;
    5) exit ;;
    *) echo "invalid option"

esac
done
```

example of for loop and basename

```
for fname in *.txt
do
   leftnm=`basename $fname  txt`
   mv $fname ${leftnm}doc
done
```