AWK PROGRAMS

13. Write an awk script to delete duplicated line from a text file. The order of the original lines must remain unchanged.

```
----------------------------------------------------------------------
BEGIN{
     count=0;
}

{
     temp = $0
     found=0
     #printf "%s\n",temp;

     for(i=1;i<=count;i++)
     {
          if(temp==arr[i])
          {
               found=1;break;
          }
     }
     if(found==0)
     {
          count++;
          arr[count]=temp
     }
}

END{
     for(i=1;i<=count;i++)
          printf "%s\n",arr[i];

}
----------------------------------------------------------------------
```

OUTPUT:

```
surbhi@ubuntu:~/Desktop/SPprac/awkProg$ cat temp13.txt
abcd efg
jklm nop
qrs
abcd efg
qrs
mnopqr

surbhi@ubuntu:~/Desktop/SPprac/awkProg$ awk -f Q13.awk temp13.txt
abcd efg
jklm nop
qrs
mnopqr

----------------------------------------------------------------------
```

14. Write an AWK Program to implement any sorting technique (Bubble/ Selction/Insertion)

BUBBLE SORT

```
BEGIN{
printf "\nSorting - BUBBLE\n"
count=0
}

{
      line[NR] = $0
      no[NR] = $1
      count++
}

END{
      for(i=1;i<=count-1;i++)
      {
            for(j=1; j<=count-i-1; j++)
            {
                  if(no[j] > no[j+1])
                  {
                        temp = line[j]
                        tempNo = no[j]
                        line[j] = line[j+1]
                        no[j] = no[j+1]
                        line[j+1] = temp
                        no[j+1] = tempNo
                  }
            }
      }

      for(i=1;i<=count;i++)
            printf "%s\n",line[i]
}
```

OUTPUT:

```
surbhi@ubuntu:~/Desktop/SPprac/awkProg$ cat temp14.txt
15    ankit rajpal
12    rajat kumar
11    arpan agrawal
9     gaurav Aggarwal
8     Saurabh agarwal
17    amit Saxena
18    rahul Saksena
surbhi@ubuntu:~/Desktop/SPprac/awkProg$ awk -f Q14_2.awk temp14.txt

Sorting - BUBBLE
8     Saurabh agarwal
```

```
9     gaurav Aggarwal
11    arpan agrawal
12    rajat kumar
15    ankit rajpal
17    amit Saxena
18    rahul Saksena

----------------------------------------------------------------------
```

SELECTION SORT

```
BEGIN{
printf "\nSorting - SELECTION\n"
count=0
}

{
      line[NR] = $0
      no[NR] = $1
      count++
}

END{
      for(i=1;i<count;i++)
      {
            minPos = i;

            for(j=i+1; j<=count; j++)
            {
                  if(no[j] < no[minPos])
                  {     minPos = j;  }
            }

            temp = line[i]
            tempNo = no[i]
            line[i] = line[minPos]
            no[i] = no[minPos]
            line[minPos] = temp
            no[minPos] = tempNo
      }


      for(i=1;i<=count;i++)
            printf "%s\n",line[i]
}
```

<u>OUTPUT:</u>

```
surbhi@ubuntu:~/Desktop/SPprac/awkProg$ cat temp14.txt
15    ankit rajpal
12    rajat kumar
11    arpan agrawal
9     gaurav Aggarwal
8     Saurabh agarwal
17    amit Saxena
18    rahul Saksena
surbhi@ubuntu:~/Desktop/SPprac/awkProg$ awk -f Q14_3.awk temp14.txt

Sorting - SELECTION
8     Saurabh agarwal
9     gaurav Aggarwal
11    arpan agrawal
```

```
12   rajat kumar
15   ankit rajpal
17   amit Saxena
18   rahul Saksena

----------------------------------------------------------------------
```

INSERTION SORT

```
BEGIN{
printf "\nSorting - INSERTION\n"
count=0
}

{
      line[NR] = $0
      no[NR] = $1
      count++
}

END{
      for(i=2;i<=count;i++)
      {
            j=i-1;
            temp = line[i];
            tempNo = no[i];

            while(j>=1 && tempNo < no[j])
            {
                  line[j+1] = line[j]
                  no[j+1] = no[j]

                  j--;

            }

            line[j+1] = temp;
      }


      for(i=1;i<=count;i++)
            printf "%s\n",line[i]
}
```

OUTPUT:

```
surbhi@ubuntu:~/Desktop/SPprac/awkProg$ cat temp14.txt
15    ankit rajpal
12    rajat kumar
11    arpan agrawal
9     gaurav Aggarwal
8     Saurabh agarwal
17    amit Saxena
18    rahul Saksena
surbhi@ubuntu:~/Desktop/SPprac/awkProg$ awk -f Q14.awk temp14.txt

Sorting - INSERTION
8     Saurabh agarwal
```

```
9    gaurav Aggarwal
11   arpan agrawal
12   rajat kumar
15   ankit rajpal
17   amit Saxena
18   rahul Saksena
----------------------------------------------------------------------
```

15. Write a shell script to check the existence of file in the current directory and folds lines of text of a file beyond 30 characters. (using awk).

```
----------------------------------------------------------------------

fold()
{
     awk '{
          len=length;
          for(i=$0; length(i)>30;)
          {
               print substr(i,1,30);

               len = len-30;
               i = substr(i,31,len);
          }
          print i;
          }' $fname
}


echo -e "\nEnter filename: "
read fname

if test -f $fname
then
     fold
else
     echo "The file does not exist"
     exit
fi

----------------------------------------------------------------------
```

OUTPUT:

```
surbhi@ubuntu:~/Desktop/SPprac/awkProg$ cat temp15.txt
Surbhi|50000aaaaaaaaaaaaaaaaaaaaaaaaaabbbbbbbbbbbbbbbbbbbbbbbbbbbjjjj
Jatin|55000dddddddddddddddddddddddddddddddddd
Harsh|65000rrrrrrrrrrrrrrrrrrrrrrrr
Yamini|60000lllllllllllllllllllllllllllllllllllllll
Deepak|55000fjfjjbrjgorrkkrrorororor
surbhi@ubuntu:~/Desktop/SPprac/awkProg$ ./Q15.sh

Enter filename:
temp15.txt
Surbhi|50000aaaaaaaaaaaaaaaaaa
aaaaaaaabbbbbbbbbbbbbbbbbbbbbb
bbbbbjjjj
Jatin|55000ddddddddddddddddddd
ddddddddddddddd
Harsh|65000rrrrrrrrrrrrrrrrrrr
rrrrr
```

```
Yamini|60000llllllllllllllllll
lllllllllllllllllllll
Deepak|55000fjfjjbrjgorrkkrror
ororor

----------------------------------------------------------------------
```

16. Write an awk script that accepts date argument in the form of mm-dd-yy and displays it in the form if day, month, and year. The script should check the validity of the argument and in case of error, display a suitable message.

```
----------------------------------------------------------------------
echo $* |
awk '
     BEGIN { FS="-" ; OFS="/" }
     {
     mm = $1
     dd = $2
     yy = $3

     if(yy < 1 || yy > 99)
     {
           print "INVALID YEAR"
           exit
     }

     if(mm < 1|| mm >12)
     {
           print "INVALID MONTH"
           exit
     }

     if(dd < 1)
     {
           print "INVALID DAY"
           exit
     }

     if(((mm==1 || mm==3 || mm==5 || mm==7 || mm==8 || mm==10 ||
mm==12)&& dd>31) || ((mm==4 || mm==6 || mm==9 || mm==11)&& dd>30) ||
(mm==2 && dd >28))
     {
           print "INVALID DAY"
           exit
     }


     #printf "Day: %s  Month: %s  Year: %s\n",dd,mm,yy
     print dd,mm,yy

      }'

----------------------------------------------------------------------
```

OUTPUT:

```
surbhi@ubuntu:~/Desktop/SPprac/awkProg$ ./Q16_2.sh 11-24-81
24/11/81
----------------------------------------------------------------------
```

17. Write a awk script that accepts two file names as arguments, checks if their permissions are identical (or) Different.

```
if [ $# -eq 2 ]
then
     if [ -f $1 -a -f $2 ]
     then
          ls -l $1 $2 | cut -d " " -f1 | tr "\n" " "  > t1.txt

          awk '$1==$2 { printf "Permissions of the two files are
same\n" }' t1.txt | tee t2.txt

          if [ ! -s t2.txt ]
          then
               echo "Persmissions are different"
          fi

          rm t1.txt t2.txt
     else
          echo "File(s) does not exist"
     fi

else
     echo "Please provide only two arguments"
fi
```

OUTPUT:

```
surbhi@ubuntu:~/Desktop/SPprac/awkProg$ ./Q17.sh temp12.txt Q16_1.sh
Persmissions are different
surbhi@ubuntu:~/Desktop/SPprac/awkProg$ ./Q17.sh Q16_2.sh  Q16_1.sh
Permissions of the two files are same
```
