CSHP 306 : Software Lab Based on CSHT 306

Ques 1) Write a shell script which accepts any number of arguments from Command line and prints them in the reverse order ..

```
:-  for n in $@
    do
     x="$n $x"
     done
    echo REVERSE ORDER IS ......
    echo $x
```

```
ubuntu@ubuntu: ~
ubuntu@ubuntu:~$ pico que1
ubuntu@ubuntu:~$ sh que1 P R I Y A N K A
REVERSE ORDER IS ......
A K N A Y I R P
ubuntu@ubuntu:~$
```

Ques 2) A unix program to eliminate multiple spaces and tbs and replace with a single space and remove empty spaces .

```
:-  echo Enter The File In Which U Want to remove White Spaces And Empty Lines ....
    read a
    echo Enter The File In Which U Want To Save The Result.....
    read b

    cat $a | tr -d "    " > $b
    cat $b | grep '\S'
```

```
ubuntu@ubuntu: ~
ubuntu@ubuntu:~$ pico que2
ubuntu@ubuntu:~$ cat abc1
Nikhil    Sukhalani



Priyanka    Sukhalani
ubuntu@ubuntu:~$ sh que2
Enter The File In Which U Want to remove White Spaces And Empty Lines ....
abc1
Enter The File In Which U Want To Save The Result.....
b
NikhilSukhalani
PriyankaSukhalani
ubuntu@ubuntu:~$
```

Ques 3) write a shell program to enhance the inbuilt cal program as below :-

a) recognise the month by name e.g. jan,Jan,JAN,January etc...
b) given zero argument ,print the current month calender
c) given one argument , print the month or year's calender
d) given two arguments , bahave like cal , except for converting month names into integers .

```
:-  echo 1. Recognise the month by name e.g. JAN,Jan,January etc
echo 2. Given zero arguments print current month calender
echo 3. Given one argument print the month or year's calender
echo 4. Given two argument print that year's corresponding month calender

case $# in
0) set `date`; m=$2 ; y=$6 ;;
1) m=$1 ; set `date`; y=$6 ;;
*) m=$1 ; y=$2 ;;
esac

case $m in
jan|Jan|January|JANUARY) m=1;;
feb|Feb|February|FEBRUARY) m=2;;
mar|Mar|march|MARCH) m=3;;
apr|Apr|april|April) m=4;;
may|May|MAY) m=5;;
june|June|JUNE) m=6;;
july|July|JULY) m=7;;
aug|Aug|august|AUGUST) m=8;;
sep|Sep|SEPTEMBER|september) m=9;;
oct|Oct|October|OCTOBER) m=10;;
nov|Nov|November|NOVEMBER) m=11;;
dec|Dec|December|DECEMBER) m=12;;
[1-9]|10|11|12) ;;
*) y=$m; m=" " ;;
esac

cal $m $y
```

```
ubuntu@ubuntu: ~
1. Recognisse the month by name e.g. JAN,Jan,January etc
2. Given zero arguments print current month calender
3. Given one argument print the month or year calender
4. Given two argument print that year corresponding monnth calender
   November 2015
Su Mo Tu We Th Fr Sa
 1  2  3  4  5  6  7
 8  9 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28
29 30

ubuntu@ubuntu:~$ sh que3 jan
1. Recognisse the month by name e.g. JAN,Jan,January etc
2. Given zero arguments print current month calender
3. Given one argument print the month or year calender
4. Given two argument print that year corresponding monnth calender
    January 2015
Su Mo Tu We Th Fr Sa
                1  2  3
 4  5  6  7  8  9 10
11 12 13 14 15 16 17
18 19 20 21 22 23 24
25 26 27 28 29 30 31

ubuntu@ubuntu:~$ sh que3 NOVEMBER
1. Recognisse the month by name e.g. JAN,Jan,January etc
2. Given zero arguments print current month calender
3. Given one argument print the month or year calender
4. Given two argument print that year corresponding monnth calender
   November 2015
Su Mo Tu We Th Fr Sa
 1  2  3  4  5  6  7
 8  9 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28
29 30

ubuntu@ubuntu:~$
```

```
1. Recognisse the month by name e.g. JAN,Jan,January etc
2. Given zero arguments print current month calender
3. Given one argument print the month or year calender
4. Given two argument print that year corresponding monnth calender
                          1996
      January               February              March
Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa
    1  2  3  4  5  6                 1  2  3                 1  2
 7  8  9 10 11 12 13   4  5  6  7  8  9 10   3  4  5  6  7  8  9
14 15 16 17 18 19 20  11 12 13 14 15 16 17  10 11 12 13 14 15 16
21 22 23 24 25 26 27  18 19 20 21 22 23 24  17 18 19 20 21 22 23
28 29 30 31           25 26 27 28 29        24 25 26 27 28 29 30
                                            31

       April                  May                   June
Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa
    1  2  3  4  5  6           1  2  3  4                    1
 7  8  9 10 11 12 13   5  6  7  8  9 10 11   2  3  4  5  6  7  8
14 15 16 17 18 19 20  12 13 14 15 16 17 18   9 10 11 12 13 14 15
21 22 23 24 25 26 27  19 20 21 22 23 24 25  16 17 18 19 20 21 22
28 29 30              26 27 28 29 30 31     23 24 25 26 27 28 29
                                            30

        July                  August               September
Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa
    1  2  3  4  5  6              1  2  3   1  2  3  4  5  6  7
 7  8  9 10 11 12 13   4  5  6  7  8  9 10   8  9 10 11 12 13 14
14 15 16 17 18 19 20  11 12 13 14 15 16 17  15 16 17 18 19 20 21
21 22 23 24 25 26 27  18 19 20 21 22 23 24  22 23 24 25 26 27 28
28 29 30 31           25 26 27 28 29 30 31  29 30


      October               November              December
Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa
       1  2  3  4  5                 1  2   1  2  3  4  5  6  7
 6  7  8  9 10 11 12   3  4  5  6  7  8  9   8  9 10 11 12 13 14
13 14 15 16 17 18 19  10 11 12 13 14 15 16  15 16 17 18 19 20 21
20 21 22 23 24 25 26  17 18 19 20 21 22 23  22 23 24 25 26 27 28
27 28 29 30 31        24 25 26 27 28 29 30  29 30 31

ubuntu@ubuntu:~$
```

Ques 4) Write a menu driven shell script to generate the following choices for user .
a) To Display the file
b) to display permissions of the file
c) To find pattern whether ignoring the case or case sensitive
d) To replace all letters 'e' by 'a'

:-

```
echo Enter the name of file .....
 read a
echo 1. Display the file
echo 2. Display Permissions of the file
echo 3. To replace all letters 'e' by 'a'
echo 4. To Find the pattern in the file
echo Enter Ur Choice :-
read ch

case "$ch" in
"1") echo The data of the file is as follows :-
    cat $a
    ;;
"2") echo The Permissions of the file is as follows :-
    set `ls -l` $a
    echo $3
    ;;
"3") echo Enter the file name in which result is to be stored .....
    read f
    tr {e} {a} <$a> $f
    cat $f
    ;;

"4") echo 1. Ignoring the case
    echo 2. Case Sensitive
    echo Enter the choice :-
    read c
    case "$c" in
    "1") echo Enter the speciifc word or letter u want to search in the file :-
        read b
        grep -i "$b" $a
        ;;
    "2") echo Enter the speciifc word or letter u want to search in the file :-
        read b
        grep "$b" $a
        ;;
         esac

esac
```

```
ubuntu@ubuntu: ~
ubuntu@ubuntu:~$ pico que4
ubuntu@ubuntu:~$ sh que4
Enter the name of file .....
abc
1. Display the file
2. Display Permissions of the file
3. To replace all letters e by a
4. To Find the pattern in the file
Enter Ur Choice :-
1
The data of the file is as follows :-
nikhil sukhalani
priyanka sukhalani
ekta sukhalani
harshita sukhalani

ubuntu@ubuntu:~$
```

```
ubuntu@ubuntu: ~

ubuntu@ubuntu:~$ sh que4
Enter the name of file .....
abc
1. Display the file
2. Display Permissions of the file
3. To replace all letters e by a
4. To Find the pattern in the file
Enter Ur Choice :-
2
The Permissions of the file is as follows :-
-rw-rw-r--
ubuntu@ubuntu:~$
```

```
ubuntu@ubuntu: ~
ubuntu@ubuntu:~$ sh que4
Enter the name of file .....
abc
1. Display the file
2. Display Permissions of the file
3. To replace all letters e by a
4. To Find the pattern in the file
Enter Ur Choice :-
3
Enter the file name in which result is to be stored .....
n
nikhil sukhalani
priyanka sukhalani
akta sukhalani
harshita sukhalani

ubuntu@ubuntu:~$
```

```
ubuntu@ubuntu: ~
ubuntu@ubuntu:~$ sh que4
Enter the name of file .....
abc
1. Display the file
2. Display Permissions of the file
3. To replace all letters e by a
4. To Find the pattern in the file
Enter Ur Choice :-
4
1. Ignoring the case
2. Case Sensitive
Enter the choice :-
1
Enter the speciifc word or letter u want to search in the file :-
LA
nikhil sukhalani
priyanka sukhalani
ekta sukhalani
harshita sukhalani
ubuntu@ubuntu:~$
```

```
ubuntu@ubuntu: ~
ubuntu@ubuntu:~$ sh que4
Enter the name of file .....
abc
1. Display the file
2. Display Permissions of the file
3. To replace all letters e by a
4. To Find the pattern in the file
Enter Ur Choice :-
4
1. Ignoring the case
2. Case Sensitive
Enter the choice :-
2
Enter the speciifc word or letter u want to search in the file :-
su
nikhil sukhalani
priyanka sukhalani
ekta sukhalani
harshita sukhalani
ubuntu@ubuntu:~$
```

Ques 5) write a menu driven shell script to generate the following choices for user :-

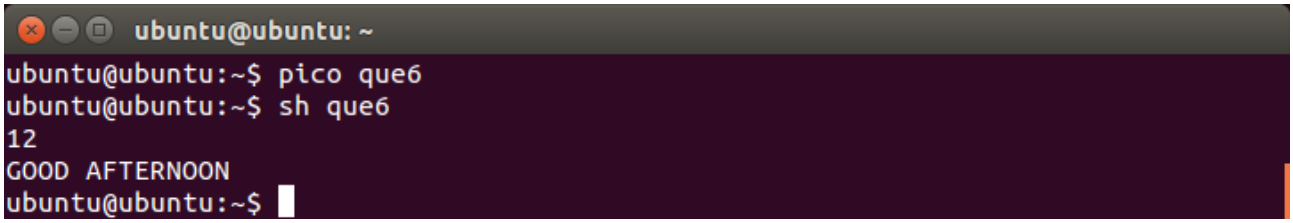
a) To display the file
b) to display the permissions of the file
c) To find the pattern in the file
    a. Ignoring the case
    b. Case sensitive
d) To replace all leters 'e' by 'a'

```
:- echo Enter the name of the file ......
read a
echo 1. To display the last n lines from the file
echo 2. To sort the file either in ascending or descending order
echo Enter the choice ....
read ch
case "$ch" in
"1") echo Enter how many last lines of the file u want to display
     read n
     tail -$n $a
     ;;
"2") echo 1. Ascending Order
     echo 2. Descending Order
     echo Enter ur choice .....
     read c
     case "$c" in
     "1")  sort $a
           ;;
     "2")  sort -r $a
           ;;
      esac
esac
```

```
  GNU nano 2.2.6                    File: que5

echo Enter the name of the file ......
read a
echo 1. To display the last n lines from the file
echo 2. To sort the file either in ascending or descending order
echo Enter the choice ....
read ch
case "$ch" in
"1") echo Enter how many last lines of the file u want to display
     read n
     tail -$n $a
     ;;

"2") echo 1. Ascending Order
     echo 2. Descending Order
     echo Enter ur choice .....
     read c
     case "$c" in
     "1")  sort $a
           ;;
     "2")  sort -r $a
           ;;
      esac
esac

                                  [ Read 23 lines ]
^G Get Help    ^O WriteOut    ^R Read File   ^Y Prev Page   ^K Cut Text    ^C Cur Pos
^X Exit        ^J Justify     ^W Where Is    ^V Next Page   ^U UnCut Text  ^T To Spell
```

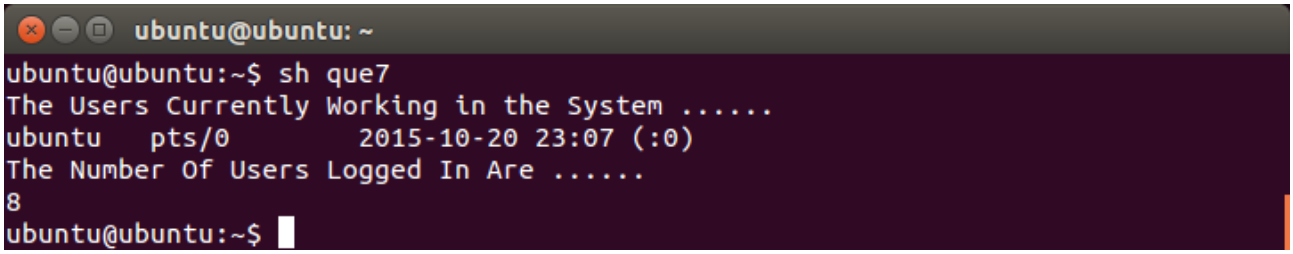
Ques 6) :- Write a shell script to print Good Morning,god Afternoon, Good Evening and Good Night ..
00:00 AM - 11:59 AM Good Morning
12:00 PM - 3:59 PM Good Afternoon
4:00 PM - 7:59 PM Good Evening
8:00 PM - 11:59 PM Good Night

```
:-  a=`date +%H`
echo $a
if [ $a -ge 0 ] && [ $a -lt 12 ]
then
  echo GOOD MORNING
elif [ $a -ge 12 ] && [ $a -lt 16 ]
then
  echo GOOD AFTERNOON
elif [ $a -ge 16 ] && [ $a -lt 20 ]
then
  echo GOOD EVENING
elif [ $a -ge 20 ] && [ $a -lt 24 ]
then
  echo GOOD NIGHT
fi
```



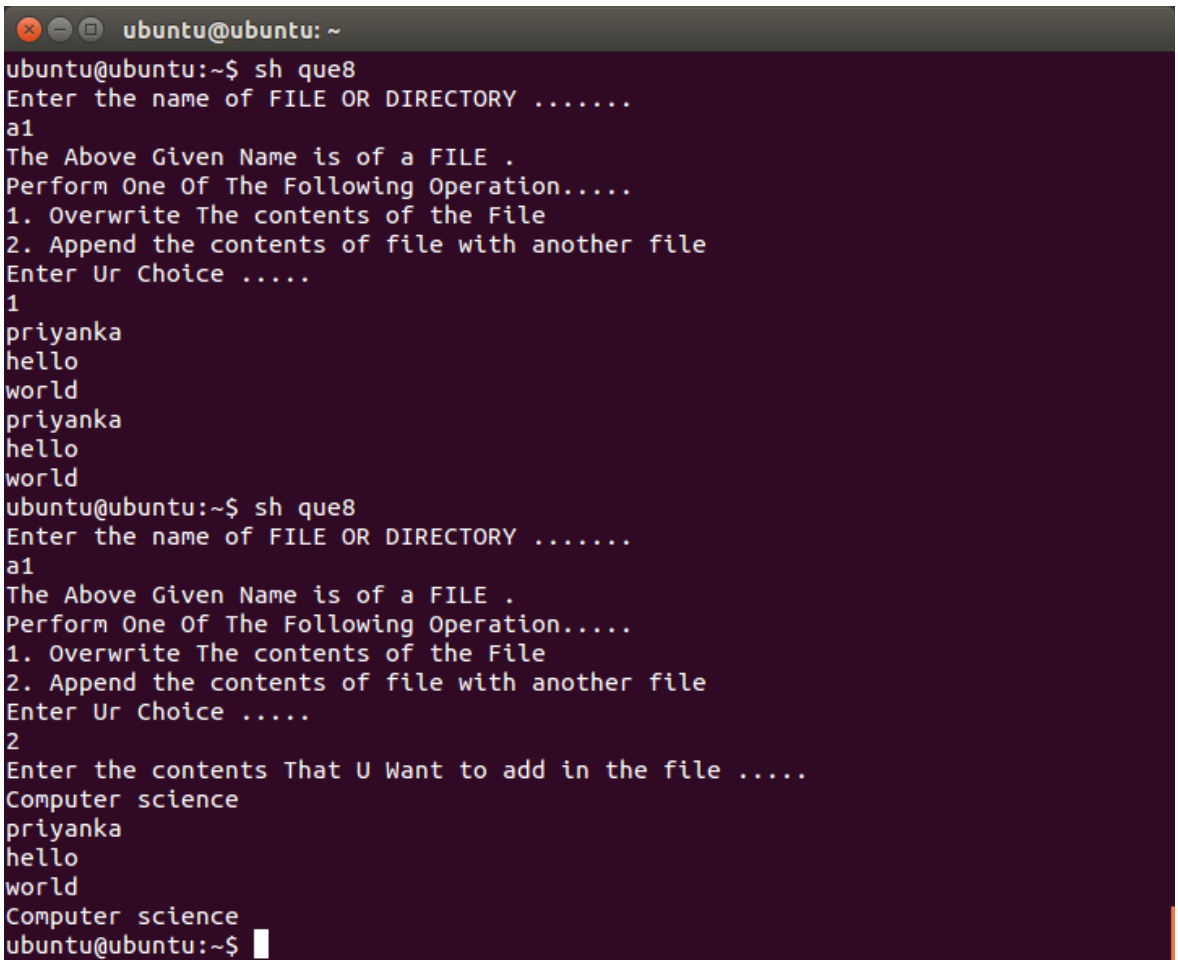
Ques 7) :- Write a shell script to list the users currents using the system along with a count of the numbers of times they have logged in

```
:-
echo The Users Currently Working in the System ......
who am i
echo The Number Of Users Logged In Are ......
who | wc -l
```



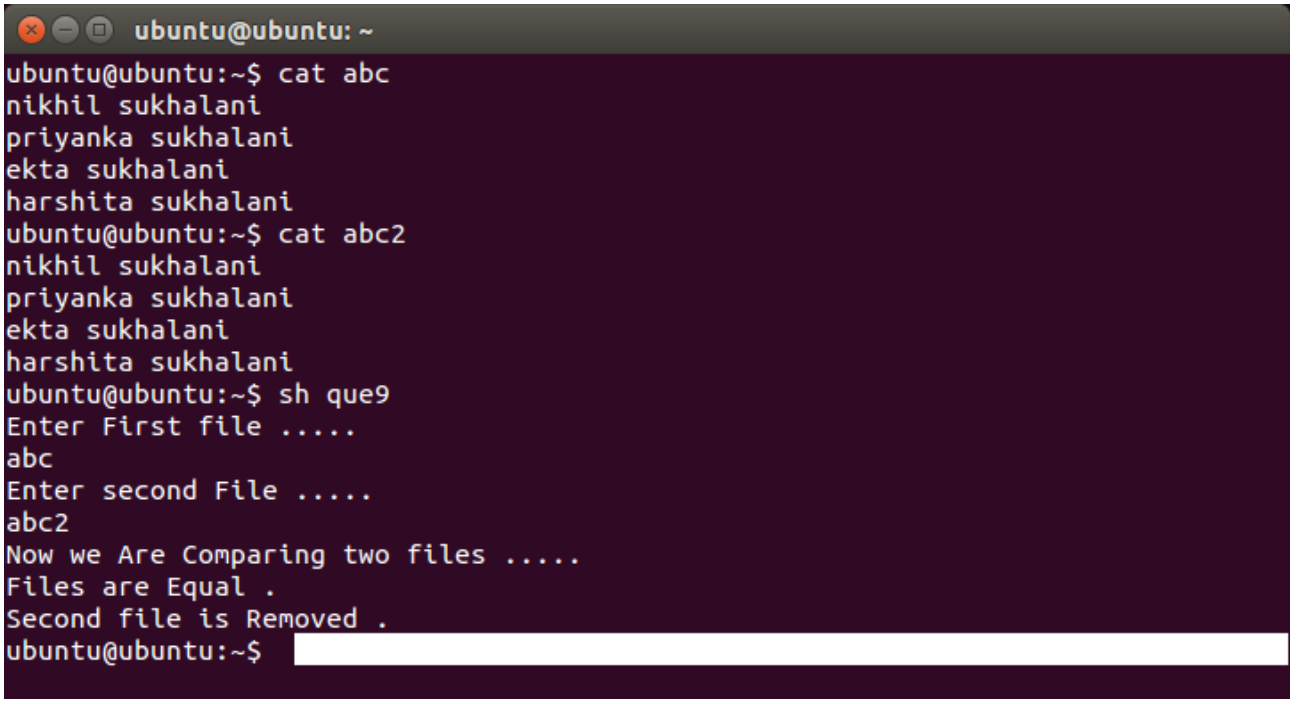
Ques 8) :- Write a shell program to accept file name or directory name from the user and only if the particular file exists not a directory, allow the user to either 1) overwrite the contents of that file or
2) append the contents in the previous contents of that file

```
:-  echo Enter the name of FILE OR DIRECTORY .......
read a
if [ -f $a ]
then
 echo The Above Given Name is of a FILE .
 echo Perform One Of The Following Operation.....
 echo 1. Overwrite The contents of the File
 echo 2. Append the contents of file with another file
 echo Enter Ur Choice .....
 read ch
 case "$ch" in
 "1") cat $a $a
      ;;
 "2") echo Enter the contents That U Want to add in the file   .....
      read b
      echo "$b" >> $a
      cat $a
      ;;
  esac
else
 echo The Above Given Name is of a DIRECTORY .

fi
```



Ques 9) Write a shell script program to compare two given file , if the contents are same remove the second one ...

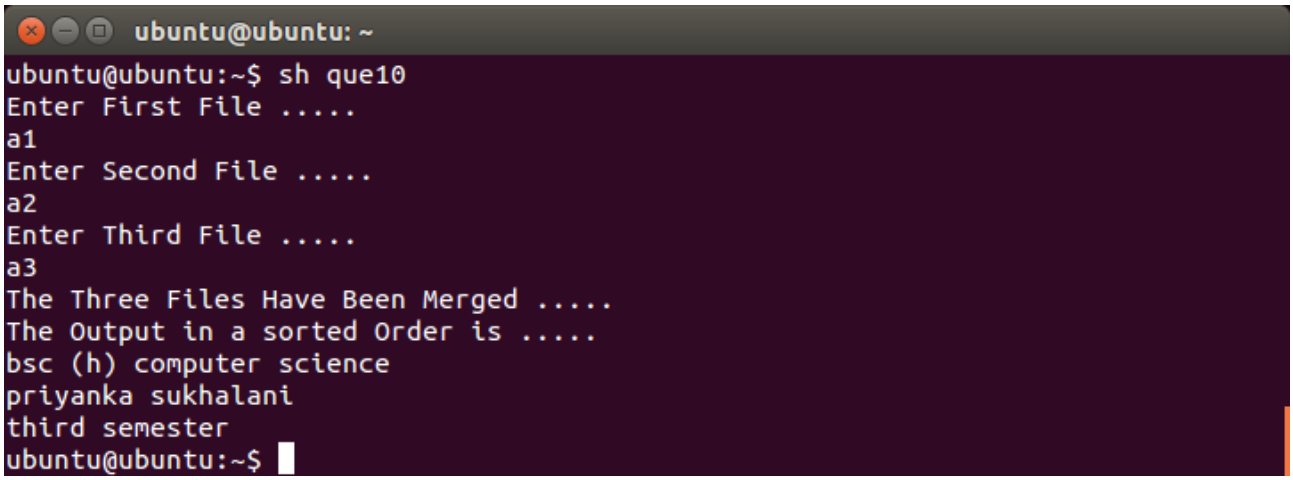
:- echo Enter First file .....
read f1
echo Enter second File .....
read f2
echo Now we Are Comparing two files .....
if cmp $f1 $f2
then
  echo Files are Equal .
  rm $f2
  echo Second file is Removed .
else
  echo Files are not Equal
fi



Ques 10) write the shell script to merge the content of three given files , sort the text contained in them and display the sorted output on the screen page by page ...

:-

```
echo Enter First File .....
read a
echo Enter Second File .....
read b
echo Enter Third File .....
read c

cat $a >> $b
cat $b >> $c
echo The Three Files Have Been Merged .....
echo The Output in a sorted Order is .....
sort $c
```



Ques 11) :- Write a shell program to write an inbuilt shell program to be able to handle following input $ cal jan mar nov ..

:- echo The Foolowing months calender is as follows :-

```
for n in $@
do
cal -m $n
done
```

```
ubuntu@ubuntu: ~
ubuntu@ubuntu:~$ pico que11
ubuntu@ubuntu:~$ sh que11 jan apr june
The Foolowing months calender is as follows :-
    January 2015
Su Mo Tu We Th Fr Sa
             1  2  3
 4  5  6  7  8  9 10
11 12 13 14 15 16 17
18 19 20 21 22 23 24
25 26 27 28 29 30 31

     April 2015
Su Mo Tu We Th Fr Sa
          1  2  3  4
 5  6  7  8  9 10 11
12 13 14 15 16 17 18
19 20 21 22 23 24 25
26 27 28 29 30

     June 2015
Su Mo Tu We Th Fr Sa
    1  2  3  4  5  6
 7  8  9 10 11 12 13
14 15 16 17 18 19 20
21 22 23 24 25 26 27
28 29 30

ubuntu@ubuntu:~$
```

Ques 12) :- Write a shell program to write an inbuilt shell program to be able to handle following input $ cal jan...nov ..

:-

```
a="$*"
set `echo $a | tr '.' ' '`
echo $1 $2

case "$1" in
jan|Jan|January|JANUARY) m=1;;
feb|Feb|February|FEBRUARY) m=2;;
mar|Mar|march|MARCH) m=3;;
apr|Apr|april|April) m=4;;
may|May|MAY) m=5;;
june|June|JUNE) m=6;;
july|July|JULY) m=7;;
aug|Aug|august|AUGUST) m=8;;
sep|Sep|SEPTEMBER|september) m=9;;
oct|Oct|October|OCTOBER) m=10;;
nov|Nov|November|NOVEMBER) m=11;;
dec|Dec|December|DECEMBER) m=12;;
esac

case "$2" in
jan|Jan|January|JANUARY) m1=1;;
feb|Feb|February|FEBRUARY) m1=2;;
mar|Mar|march|MARCH) m1=3;;
apr|Apr|april|April) m1=4;;
may|May|MAY) m1=5;;
june|June|JUNE) m1=6;;
july|July|JULY) m1=7;;
aug|Aug|august|AUGUST) m1=8;;
sep|Sep|SEPTEMBER|september) m1=9;;
oct|Oct|October|OCTOBER) m1=10;;
nov|Nov|November|NOVEMBER) m1=11;;
dec|Dec|December|DECEMBER) m1=12;;
esac

while [ $m -le $m1 ]
do
cal $m 2015
m=`expr $m + 1`
done
```

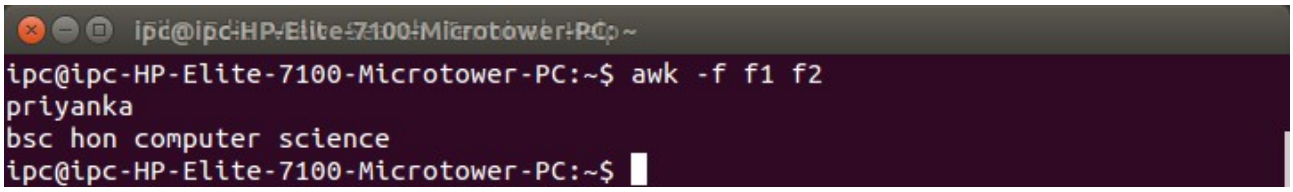
```
ubuntu@ubuntu: ~
  GNU nano 2.2.6                File: que12

nov|Nov|November|NOVEMBER) m=11;;
dec|Dec|December|DECEMBER) m=12;;
esac

case "$2" in
jan|Jan|January|JANUARY) m1=1;;
feb|Feb|February|FEBRUARY) m1=2;;
mar|Mar|march|MARCH) m1=3;;
apr|Apr|april|April) m1=4;;
may|May|MAY) m1=5;;
june|June|JUNE) m1=6;;
july|July|JULY) m1=7;;
aug|Aug|august|AUGUST) m1=8;;
sep|Sep|SEPTEMBER|september) m1=9;;
oct|Oct|October|OCTOBER) m1=10;;
nov|Nov|November|NOVEMBER) m1=11;;
dec|Dec|December|DECEMBER) m1=12;;
esac

while [ $m -le $m1 ]
do
cal $m 2015
m=`expr $m + 1`
done

^G Get Help  ^O WriteOut  ^R Read File ^Y Prev Page ^K Cut Text  ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is  ^V Next Page ^U UnCut Text^T To Spell
```

```
ubuntu@ubuntu: ~

ubuntu@ubuntu:~$ pico que11
ubuntu@ubuntu:~$ pico que12
ubuntu@ubuntu:~$ sh que12 jan...sep
jan sep
    January 2015
Su Mo Tu We Th Fr Sa
              1  2  3
 4  5  6  7  8  9 10
11 12 13 14 15 16 17
18 19 20 21 22 23 24
25 26 27 28 29 30 31

   February 2015
Su Mo Tu We Th Fr Sa
 1  2  3  4  5  6  7
 8  9 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28


     March 2015
Su Mo Tu We Th Fr Sa
 1  2  3  4  5  6  7
 8  9 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28
29 30 31

     April 2015
Su Mo Tu We Th Fr Sa
          1  2  3  4
 5  6  7  8  9 10 11
12 13 14 15 16 17 18
19 20 21 22 23 24 25
26 27 28 29 30

      May 2015
Su Mo Tu We Th Fr Sa
```

```
ubuntu@ubuntu: ~
                 1  2
 3  4  5  6  7  8  9
10 11 12 13 14 15 16
17 18 19 20 21 22 23
24 25 26 27 28 29 30
31
     June 2015
Su Mo Tu We Th Fr Sa
    1  2  3  4  5  6
 7  8  9 10 11 12 13
14 15 16 17 18 19 20
21 22 23 24 25 26 27
28 29 30

     July 2015
Su Mo Tu We Th Fr Sa
          1  2  3  4
 5  6  7  8  9 10 11
12 13 14 15 16 17 18
19 20 21 22 23 24 25
26 27 28 29 30 31

    August 2015
Su Mo Tu We Th Fr Sa
                   1
 2  3  4  5  6  7  8
 9 10 11 12 13 14 15
16 17 18 19 20 21 22
23 24 25 26 27 28 29
30 31
   September 2015
Su Mo Tu We Th Fr Sa
       1  2  3  4  5
 6  7  8  9 10 11 12
13 14 15 16 17 18 19
20 21 22 23 24 25 26
27 28 29 30

ubuntu@ubuntu:~$
```

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

:- {
Arr[n++]=$0}
END{
for(i=0;i<n;i++)
{
flag=0;
for(j=0;j<i;j++)
{
if(Arr[i] == Arr[j])
{
flag=1;
break;
}
}
if(flag == 0)
print Arr[i]
}
}



Ques 14) Write an awk program to implement any sorting technique .

:-
{
ar[NR] = $0
}
{
for(i=0;i<=NR;i++)
{
for(j=i+1;j<=NR-1;j++)
{
if(ar[j] > ar[j+1])
{
t = ar[j];
ar[j] = ar[j+1]
ar[j+1] = t
}
}
}
}
END{ for(i=1; i<=NR ; i++)
  { print ar[i]
} }

```
ipc@ipc-HP-Elite-7100-Microtower-PC:~$ pico f4
ipc@ipc-HP-Elite-7100-Microtower-PC:~$ awk -f f3 f4
1
2

3
5
8
ipc@ipc-HP-Elite-7100-Microtower-PC:~$
```

Ques 15) :- Write a shell script to check the existence of file in the current directory and folds line of text of file beyond 30 characters

:-

```
END{
for(i=1;i<length;i+=30)
print substr($0,i,30)
}
```

```
ipc@ipc-HP-Elite-7100-Microtower-PC:~$ pico f5
ipc@ipc-HP-Elite-7100-Microtower-PC:~$ pico f6
ipc@ipc-HP-Elite-7100-Microtower-PC:~$ awk -f f5 f6
priyanka sukhalani harshita su
khalani
ipc@ipc-HP-Elite-7100-Microtower-PC:~$
```

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

:-

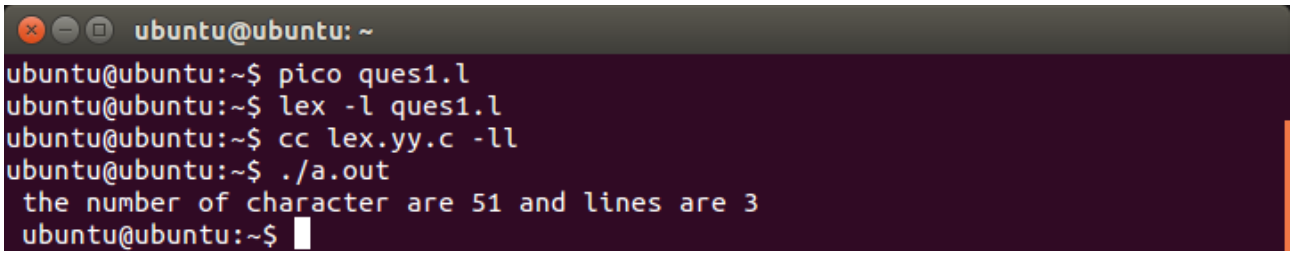
```
BEGIN {
FS="-"
f=1;
print "Enter date with (mm-dd-yy) format :- ";
getline "/dev/tty"
print $1;
print $2;
print $3;
if((($3%4!=0) && ($1==2) && ($2>28)) || (($3%4==0) && ($1==2) && ($2>29))) ||
(($1==1) || ($1==3) || ($1==5) || ($1==7) || ($1==8) || ($1==10) || ($1==12)
&& ($2>31)) ||
(($1==4) || ($1==6) || ($1==9) || ($1==11) && ($2>30)) || ($1<1) || ($2<1) ||
($3<1) || ($1>12))
f=0;
if(f==0)
```

```
print "You Have Entered an invalid Date ... ";
else
{
print " The date = " $2;
print " The month = " $1;
print " The Year = " $3;
print " is valid date ";
}
}
```

LEX PROGRAMMING

Ques 1) Write a lex program to count the no of lines and characters in the input file .

```
:- %{
int line=0, ch=0;
%}
%%
\n {line++;}
. {ch++;}
%%
main()
{
yyin=fopen("n.txt","r");
yylex();
printf(" the number of character are %d and lines are %d \n ",ch,line);
}
```



Ques 2) Write a lex program that implements the ceaser cipher : it replace every letter with one three letter after in alphabetical order, wrapping around at Z .

```
:- %%
[a-z]    { char ch = yytext[0];
         ch += 3;
         if (ch > 'z') ch -= ('z'+1-'a');
         printf ("%c", ch);
        }

[A-Z]   { char ch = yytext[0];
```

```
        ch += 3;
        if (ch > 'Z') ch -= ('Z'+1-'A');
        printf ("%c", ch);
      }
%%
int main()
{
yyin=fopen("new","r");
yylex();
}
```

```
ubuntu@ubuntu: ~
ubuntu@ubuntu:~$ pico ques2.l
ubuntu@ubuntu:~$ lex -l ques2.l
ubuntu@ubuntu:~$ cc lex.yy.c -ll
ubuntu@ubuntu:~$ ./a.out
sulbdqnd vxnkdodql
evf krq frpsxwhu vflhqfh
wklug vhp
ubuntu@ubuntu:~$
```

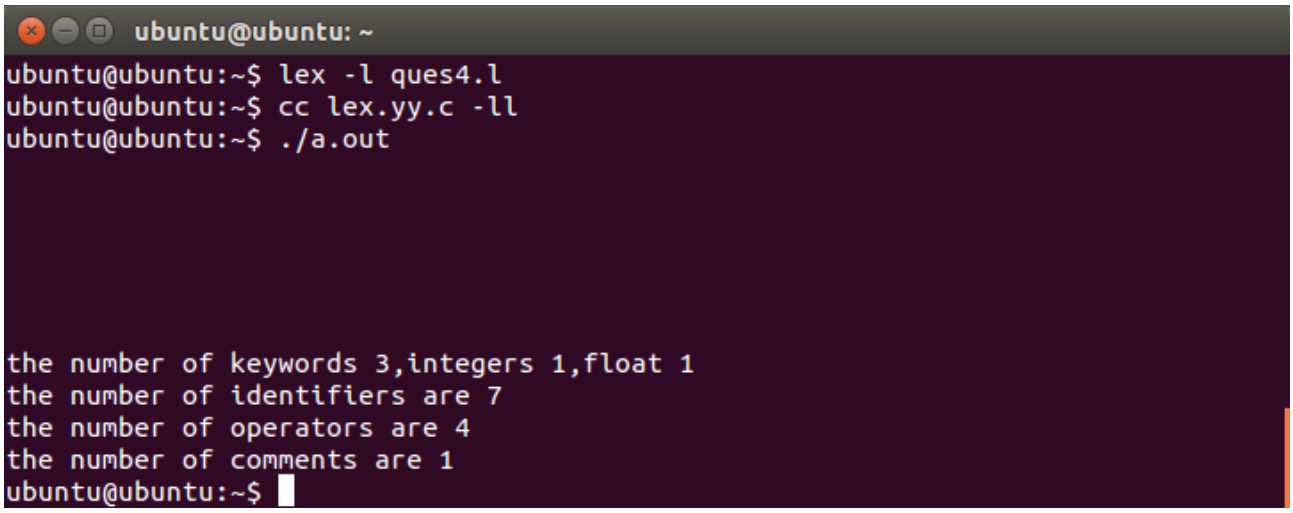
Ques 3) :- Write a lex program that find the longest word in the input .

```
:-  %{
#include <strings.h>
int longest = 0;
char longword[60];
%}

%%
[a-zA-Z]+       { if (yyleng > longest) {
                  longest = yyleng;
                  strcpy (longword, yytext);
                  }
              }
.         |
\n        ;
%%
int main () {
   yyin=fopen("abc.txt","r");
   yylex ();
   printf ("The longest word was \"%s\", which was %d characters long.\n",
            longword, longest);
   return 0;
   }
```

```
ubuntu@ubuntu: ~
ubuntu@ubuntu:~$ pico ques3.l
ubuntu@ubuntu:~$ pico ques3.l
ubuntu@ubuntu:~$ lex -l ques3.l
ubuntu@ubuntu:~$ cc lex.yy.c -ll
ubuntu@ubuntu:~$ ./a.out
The longest word was "Sukhalani", which was 9 characters long.
ubuntu@ubuntu:~$
```

Ques4) :- Write a lex program that ditinguishes integer ,keyword ,identifiers,floats,operators and comments in any simple programming language .

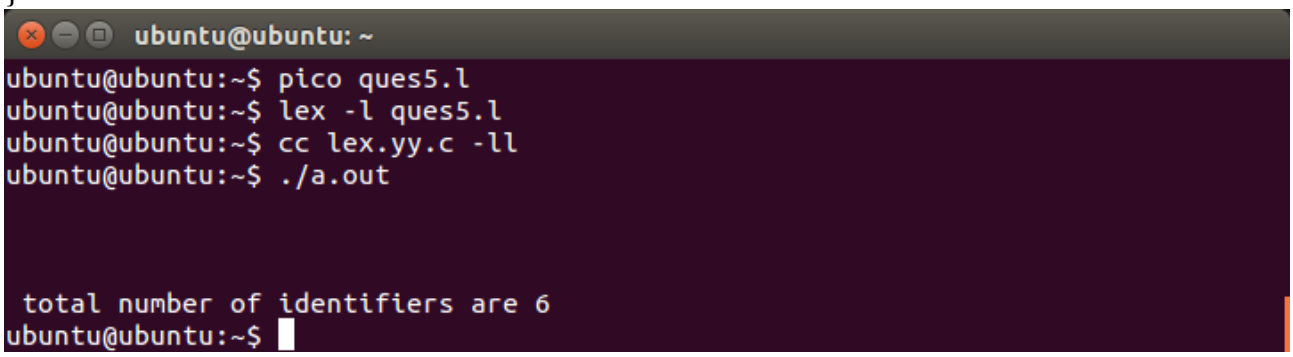
```
:- %{
int key=0,in=0,fl=0,id=0,op=0,co=0;
%}
%%
int|scanf|printf|main {key++;}
[0-9]* {in++;}
[0-9]+"."[0-9]* {fl++;}
[a-z]*[0-9]* {id++;}
"+"|"-"|"*"|"/"|"="|"()"|";" {op++;}
"//"|"/*" {co++;}
%%
main()
{
yyin=fopen("c++.txt","r");
yylex();
printf("the number of keywords %d,integers %d,float %d\n",key,in,fl);
printf("the number of identifiers are %d\n",id);
printf("the number of operators are %d\n",op);
printf("the number of comments are %d\n",co);
}
```



Ques 5) :- Write a lex program to count the no of identifiers in a C file .

```
:-
%{
int id_cnt=0;
char ch;
%}
%%
"int"|"float"|"double"|"char"  { ch=input();
for(;;)
{
if(ch==',')
```

```
      id_cnt++;
else if(ch==';')
      break;
ch=input();
}
id_cnt++;
}
%%
main(int argc,char **argv)
{
FILE *fp;
yyin=fopen("mn2.txt","r");
yylex();
printf("\n total number of identifiers are %d",id_cnt);
}
```



Ques 6) :- Write a lex program to count the no of word , character, blank spaces and lines in a C file .

```
:- %{
int word=0,lines=0,ch=0,space=0;
%}

%%
[\t' ']     {word++;space++;}
[\n' ']     {lines++;word++;}
[a-z]|[A-Z]  {ch++;}
%%

main()
{
 yyin=fopen("mn","r");
 yylex();
 printf("ToTal Spaces %d \n",space);
 printf("ToTal Words %d \n",word);
 printf("ToTal Characters %d \n",ch);
 printf("ToTal Lines %d \n",lines);
}
```

```
ubuntu@ubuntu: ~
ubuntu@ubuntu:~$ pico que6.l
ubuntu@ubuntu:~$ lex -l que6.l
ubuntu@ubuntu:~$ cc lex.yy.c -ll
ubuntu@ubuntu:~$ ./a.out
ToTal Spaces 5
ToTal Words 8
ToTal Characters 46
ToTal Lines 3
ubuntu@ubuntu:~$
```

Ques 7) :- write a lex specification program that generates a C program which takes a string
abcd
abc
ab
a

```
:- %{
%}

%%
a|ab|abc|abcd printf("%s\n",yytext);REJECT
.|\n
%%

main()
{
printf("\n Enter the data ... :- ");
yylex();
return 0;
}
```

```
ubuntu@ubuntu: ~
ubuntu@ubuntu:~$ pico ques7.l
ubuntu@ubuntu:~$ lex -l ques7.l
ubuntu@ubuntu:~$ cc lex.yy.c -ll
ubuntu@ubuntu:~$ ./a.out

 Enter the data ... :- abcd
abcd
abc
ab
a
```

Ques 8) :- A program in lex to recognise a valid arithmetic expression .

```
:- %{
#include<stdio.h>
int a=0,b=0,c=0,d=0,ob=0,cb=0;
int flaga=0,flagb=0,flagc=0,flagd=0;
%}
```

```
%%
[a-zA-z]+ printf("\n %s is an identifier\n",yytext);
[+] {a++;flaga=1;}
[-] {b++;flagb=1;}
[*] {c++;flagc=1;}
[/] {d++;flagd=1;}
[(] ob++;
[)] cb++;
%%
main()
{
printf("Enter expression:");
yylex();
if(ob==cb)
printf("\nvalid expression\n");
else
printf("invalid expression\n");
printf("Addition=%d\tSubtract=%d\nMultiply=%d\tDivide=%d\n",a,b,c,d);
printf("\n Operators used :- ");
if(flaga==1)
printf("+\t");
if (flagb==1)
printf("-\t");
if(flagc==1)
printf("*\t");
if(flagd==1)
printf("/\n");
}
yywrap(){}
yyerror(){}
```

```
ubuntu@ubuntu: ~
ubuntu@ubuntu:~$ pico ques8.l
ubuntu@ubuntu:~$ lex -l ques8.l
ubuntu@ubuntu:~$ cc lex.yy.c -ll
ubuntu@ubuntu:~$ ./a.out
Enter expression:a=(b+c)*d

 a is an identifier
=
 b is an identifier

 c is an identifier

 d is an identifier


valid expression
Addition=1      Subtract=0
Multiply=1      Divide=0

 Operators used :- +    *
ubuntu@ubuntu:~$
```

THE END....

Made By
Nehal Pandey
Bsc(H)Computer Science
III Semester
Indraprastha College for Women