SHELL PROGRAMMING

1. Write a shell script which accepts any number of arguments from command line and prints them in the reverse order(for example, if script is named rags, then executing rags A B C should produce C B A on the standard output). (use for loop).

```
---------------------------------------------------------------------
argv=$*
argc=$#

rev=""

for((i=1; $i<=$argc; i++))
do
      ((n=$argc-$i))
      shift $n
      rev="$rev $1"
      set $argv
done

echo $rev
---------------------------------------------------------------------
```

OUTPUT:

```
surbhi@ubuntu:~/Desktop/SPprac/programs$ ./Q1_2.sh abc def ghi
ghi def abc
---------------------------------------------------------------------
```

2. A Unix program to eliminate multiple spaces and tabs and replace with a single space and remove empty lines.

```
echo -e "\nEnter filename: "
read fname

if test -f $fname
then
echo "Removing spaces";
else
echo "The file doesn't exist. Exiting...";
exit
fi

cat $fname | tr '\t' ' ' > temp1
cat temp1 | tr -s ' \n' > temp2

rm temp1
rm $fname
mv temp2 $fname

echo -e "\nFile after removing spaces :- "
cat $fname
```

OUTPUT:

```
surbhi@ubuntu:~/Desktop/SPprac/programs$ cat > new.txt
ajs
ffj       jj

4fkkf   ll

rr


surbhi@ubuntu:~/Desktop/SPprac/programs$ ./Q2.sh

Enter filename:
new.txt
Removing spaces

File after removing spaces :-
ajs
ffj jj
4fkkf ll
rr
```

---

3. Write a Shell program to enhance the inbuilt cal program as below:

- **Recognize the month by name. e.g. jan, Jan, JAN, January etc.**
- **Given zero arguments, should print the current month's calendar.**
- **Given one argument, prints the month or year's calendar.**
- **Given two arguments, should behave like cal, except for converting month names into numbers.**

```
argc=$#

if [ $argc -gt 2 ]
then
echo -e "\nMaximum no of arguments are 2\n";
exit
fi

if [ $argc -eq 0 ]
then
month=`date|cut -d " " -f2`
year=`date | cut -d " " -f6`
elif [ $argc -eq 1 ]
then
      case $1 in
      [1-9]*) year=$1
            month=0 ;;
      *)    month=$1
            year=`date | cut -d " " -f6` ;;
      esac
else
month=$1
year=$2
fi

month=`echo $month | tr [:lower:] [:upper:]`

case $month in
JAN*) month=1;;
FEB*) month=2;;
MAR*) month=3 ;;
APR*) month=4;;
MAY*) month=5 ;;
JUN*) month=6 ;;
JUL*) month=7 ;;
AUG*) month=8 ;;
SEP*) month=9;;
OCT*) month=10 ;;
NOV*) month=11 ;;
DEC*) month=12 ;;
esac
```

```
      if [ $month -eq 0 ]
      then
      cal $year
      else
      cal $month $year
      fi
```

OUTPUT:

```
surbhi@ubuntu:~/Desktop/SPprac/programs$ ./Q3_2.sh feb
   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


surbhi@ubuntu:~/Desktop/SPprac/programs$ ./Q3_2.sh feb 2014
   February 2014
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
```

4. Write a menu driven shell script to generate the following choices for user:
i) To display the file.
ii) To display the permissions of the file.
iii) To find the pattern in the file
a. ignoring the case and
b. case sensitive. (using grep)
iv) To replace all letters ‘e’ by ‘a’.

```
menu()
{
      echo "------------------------------------------------"
      echo -e "File Options"
      echo -e "1.Display the file."
      echo -e "2.Display the permissions of the file."
      echo -e "3.Find a pattern in the file."
      echo -e "4.Replace all letters 'e' by 'a'"
      echo "------------------------------------------------"
      echo -e "Enter your choice: "
      echo "------------------------------------------------"
      read ch
      echo "------------------------------------------------"

      choice
}

choice()
{
      case $ch in

      1) echo "Your file is-"
         echo "------------------------------------------------"
         cat $fname;;

      2) echo -e "\nThe permissions of the file are-"
            ls -l $fname | cut -d " " -f1,9;;

      3) echo -e "\nEnter the pattern to be searched-"
         read patt

         echo "------------------------------------------------"
         echo -e "    1.Ignoring case"
         echo -e "    2.Case sensitive"
         echo "------------------------------------------------"
         echo -ne "   Enter your choice: "
         read ch1
         echo "------------------------------------------------"

         case $ch1 in
```

```
         1)  grep -i $patt $fname

           if [ $? -ne 0 ]
           then
                echo "Pattern not found."
           fi ;;

         2) grep $patt $fname

           if [ $? -ne 0 ]
           then
                echo "Pattern not found."
           fi ;;

         *) echo -e "\nPlease enter a valid choice (1 or 2)"
         esac ;;


      4) cat $fname | tr 'e' 'a' ;;

      *) echo -e "\nPlease enter a valid choice"

      esac

      continueYN
}

continueYN()
{
      echo "-------------------------------------------------"
      echo -e -n "\nDo you wish to continue? : "
      read c

      if [ $c = 'y' ]
      then
      menu
      else
      exit
      fi
}

echo "-------------------------------------------------"
echo -e "Enter a filename"
read fname

if test -e $fname
then
menu
else
echo -e "\nThe file does not exist. Please try again. "
fi
-----------------------------------------------------------------------
```

OUTPUT:

```
------------------------------------------------
Enter a filename
new.txt
------------------------------------------------
File Options
1.Display the file.
2.Display the permissions of the file.
3.Find a pattern in the file.
4.Replace all letters 'e' by 'a'
------------------------------------------------
Enter your choice:
------------------------------------------------
1
------------------------------------------------
Your file is-
------------------------------------------------
ajs
ffj jj
4fkkf ll
rr
------------------------------------------------

Do you wish to continue? : y
------------------------------------------------
File Options
1.Display the file.
2.Display the permissions of the file.
3.Find a pattern in the file.
4.Replace all letters 'e' by 'a'
------------------------------------------------
Enter your choice:
------------------------------------------------
2
------------------------------------------------

The permissions of the file are-
-rw-rw-r-- new.txt
------------------------------------------------

Do you wish to continue? : y
------------------------------------------------
File Options
1.Display the file.
2.Display the permissions of the file.
3.Find a pattern in the file.
4.Replace all letters 'e' by 'a'
------------------------------------------------
Enter your choice:
------------------------------------------------
```

```
3
--------------------------------------------------

Enter the pattern to be searched-
efg
--------------------------------------------------
     1.Ignoring case
     2.Case sensitive
--------------------------------------------------
     Enter your choice: 2
--------------------------------------------------
Pattern not found.
--------------------------------------------------

Do you wish to continue? : n
```

5. Write a menu-driven shell script to generate the following choices for user:
a) To display the last n (entered by user) lines from the file.
b) To sort the file in either ascending order or descending order. (using sort command)

```
menu()
{
      echo "------------------------------------------------"
      echo -e "File Options"
      echo -e "1.Display the file."
      echo -e "2.Display last n lines of the file."
      echo -e "3.Sort the file."
      echo "------------------------------------------------"
      echo -ne "Enter your choice: "
      read ch
      echo "------------------------------------------------"

      choice
}

choice()
{
      case $ch in

      1) echo "Your file is-"
         echo "------------------------------------------------"
         cat $fname;;

      2) echo -ne "\nEnter n: "
         read n
         echo "------------------------------------------------"
         tail -$n $fname  ;;

      3) echo -e "    1.Ascending order."
         echo -e "    2.Descending order."
         echo "------------------------------------------------"
         echo -ne "   Enter your choice: "
         read ch1
         echo "------------------------------------------------"

         case $ch1 in

         1) sort $fname   ;;

         2) sort -r $fname ;;

         *) echo -e "\nPlease enter a valid choice ( 1 or 2)"
         esac ;;
```

```
      *) echo -e "\nPlease enter a valid choice"
         esac

      continueYN
}

continueYN()
{
      echo "-------------------------------------------------"
      echo -e -n "\nDo you wish to continue? : "
      read c

      if [ $c = 'y' ]
      then
      menu
      else
      exit
      fi
}

echo "-------------------------------------------------"
echo -e "Enter a filename"
read fname

if test -e $fname
then
menu
else
echo -e "\nThe file does not exist. Please try again. "
fi
```

OUTPUT:

```
-------------------------------------------------
Enter a filename
new.txt
-------------------------------------------------
File Options
1.Display the file.
2.Display last n lines of the file.
3.Sort the file.
-------------------------------------------------
Enter your choice: 2
-------------------------------------------------

Enter n: 2
-------------------------------------------------
4fkkf ll
rr
-------------------------------------------------
```

```
Do you wish to continue? : y
--------------------------------------------------
File Options
1.Display the file.
2.Display last n lines of the file.
3.Sort the file.
--------------------------------------------------
Enter your choice: 3
--------------------------------------------------
     1.Ascending order.
     2.Descending order.
--------------------------------------------------
     Enter your choice: 1
--------------------------------------------------
4fkkf ll
ajs
ffj jj
rr
--------------------------------------------------

Do you wish to continue? : n
```

6. Write a shell script to print the Good morning, Good Afternoon, Good Evening or Good Night according to the time of the day.
00:00 A.M – 11:59 A.M: Good Morning
12:00 P.M -3:59 P.M: Good Afternoon
4:00 P.M-7:59 P.M: Good Evening
8:00 P.M-11:59: Good Night

```
hour=`date | cut -d " " -f4 | cut -d ":" -f1`

case $hour in
[0-1][0-1]|0[2-9] ) echo -e "\nGood Morning" ;;

1[2-5]) echo -e "\nGood Afternoon" ;;

1[6-9]) echo -e "\nGood Evening" ;;

*) echo -e "\nGood Night";;

esac

echo ""
```

OUTPUT:

```
surbhi@ubuntu:~/Desktop/SPprac/programs$ date
Sun Aug 30 16:20:48 IST 2015
surbhi@ubuntu:~/Desktop/SPprac/programs$ ./Q6_1.sh

Good Evening
```

7. Write a shell script to list the users currently using the system along with a count of the numbers of times they have logged in.

```
who | cut -d " " -f1 | sort | uniq > temp1
cat temp1 | tr "\n" " " > temp2
((count=`cat temp1 | wc -l`))

for((i=1;i<=count;i++))
do
      patt=`cat temp2 | cut -d " " -f$i`
      uniqueCount=`who | grep -cw "$patt"`
      echo "$patt is logged in $uniqueCount times"
done
```

OUTPUT:

```
surbhi@ubuntu:~/Desktop/SPprac/programs$ who
surbhi   :0           2015-08-30 15:05 (:0)
surbhi   pts/1        2015-08-30 16:02 (:0)
surbhi@ubuntu:~/Desktop/SPprac/programs$ ./Q7.sh
surbhi is logged in 2 times
```

8. Write a Shell program to accept filename or Directory name from the user and only if the particular file exists and not a directory, allow the user to either i) overwrite the contents of that file or ii) append the contents in the previous contents of that file.

```
menu()
{
      echo "------------------------------------------------"
      echo -e "File Options"
      echo -e "1.Display the file."
      echo -e "2.Overwrite the contents of the file."
      echo -e "3.Append data to the file."
      echo "------------------------------------------------"
      echo -ne "Enter your choice: "
      read ch
      echo "------------------------------------------------"

      choice
}

choice()
{
      case $ch in

      1) echo "Your file is-"
         echo "------------------------------------------------"
         cat $fname;;

      2) echo "(Press Ctrl+D to save file)"
         cat > $fname ;;

      3) echo "(Press Ctrl+D to save file)"
         cat >> $fname ;;

      *) echo -e "\nPlease enter a valid choice"

      esac

      continueYN
}

continueYN()
{
      echo "------------------------------------------------"
      echo -e -n "Do you wish to continue? : "
      read c

      if [ $c = 'y' ]
      then
      menu
```

```
      else
      exit
      fi
}

echo "------------------------------------------------"
echo -e "Enter a filename"
read fname

if test -e $fname
then
      if test -d $fname
      then
      echo "------------------------------------------------"
      echo "$fname is a directory"
      echo "------------------------------------------------"
      else
      menu
      fi
else
echo "------------------------------------------------"
echo -e "\nThe file does not exist. Please try again. "
echo "------------------------------------------------"
fi
```

OUTPUT:

```
------------------------------------------------
Enter a filename
new.txt
------------------------------------------------
File Options
1.Display the file.
2.Overwrite the contents of the file.
3.Append data to the file.
------------------------------------------------
Enter your choice: 1
------------------------------------------------
Your file is-
------------------------------------------------
ajs
ffj jj
4fkkf ll
rr
------------------------------------------------
Do you wish to continue? : y
------------------------------------------------
File Options
1.Display the file.
2.Overwrite the contents of the file.
3.Append data to the file.
```

```
--------------------------------------------------
Enter your choice: 3
--------------------------------------------------
(Press Ctrl+D to save file)
abd
mkl
--------------------------------------------------
Do you wish to continue? : y
--------------------------------------------------
File Options
1.Display the file.
2.Overwrite the contents of the file.
3.Append data to the file.
--------------------------------------------------
Enter your choice: 1
--------------------------------------------------
Your file is-
--------------------------------------------------
ajs
ffj jj
4fkkf ll
rr
abd
mkl
--------------------------------------------------
Do you wish to continue? :
```

9. Write the shell script to compare two given files, if the contents are same remove the second one.

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

compare()
{
      echo ""
      cmp $fname1 $fname2
      val=`echo $?`

      if [ $val -eq 0 ]     #the files are same
      then
            echo -e "\nThe files are same."
            rm $fname2
            echo "$fname2 has been deleted."
      else
      echo -e "\nThe files are different. No action required."
      fi

      echo "-------------------------------------------------"
}


echo "-------------------------------------------------"
echo -e "Enter first filename"
read fname1

echo -e "\nEnter second filename"
read fname2

if [ -e $fname1 -a -e $fname2 ]
then
      if [ "$fname1" = "$fname2" ]
      then
      echo -e "\nBoth files have the same name. Invalid input."
      echo "-------------------------------------------------"
      else
      compare
      fi
else
echo -e "\nOne or both files do not exist. Please try again. "
echo "-------------------------------------------------"
fi

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

OUTPUT:

```
surbhi@ubuntu:~/Desktop/SPprac/programs$ ./Q9.sh
-------------------------------------------------
Enter first filename
Q9f1.txt

Enter second filename
Q9f2.txt

cmp: EOF on Q9f1.txt

The files are different. No action required.
-------------------------------------------------
```

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

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

mergeAndSort()
{
      echo "------------------------------------------------"
      sort -m $fname1 $fname2 $fname3 > merge
      echo "Merged and sorted file :-"
      echo "------------------------------------------------"
      sort merge | more
      rm merge
}




echo "------------------------------------------------"
echo -e "Enter first filename"
read fname1

if test -e $fname1
then
      echo -e "\nEnter second filename"
      read fname2

      if test -e $fname2
      then
            echo -e "\nEnter third filename"
            read fname3

            if test -e $fname3
            then
            mergeAndSort
            else
            echo -e "\nThis file does not exist. Please try again. "
            fi
      else
      echo -e "\nThis file does not exist. Please try again. "
      fi
else
echo -e "\nThis file does not exist. Please try again. "
fi
echo "------------------------------------------------"

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

OUTPUT:

```
--------------------------------------------------
Enter first filename
new.txt

Enter second filename
Q9f1.txt

Enter third filename
Q9f2.txt
--------------------------------------------------
Merged and sorted file :-
--------------------------------------------------
4fkkf ll
abc
abc
abd
ajs
def
def
def
ffj jj
ghi
jkl
lmn
mkl
rr
--------------------------------------------------
```

11. Write a Shell program to modify the inbuilt cal program to be able to handle following input:

$ cal jan mar nov

```
argc=$#

if [ $argc -eq 0 ]
then
echo -e "\nAt least one argument is required\n"
exit
fi

year=`date | cut -d " " -f6`

for arg in $*
{
      arg=`echo $arg | tr [:lower:] [:upper:]`

      case $arg in
      JAN*) val=1;;
      FEB*) val=2;;
      MAR*) val=3 ;;
      APR*) val=4;;
      MAY*) val=5 ;;
      JUN*) val=6 ;;
      JUL*) val=7 ;;
      AUG*) val=8 ;;
      SEP*) val=9;;
      OCT*) val=10 ;;
      NOV*) val=11 ;;
      DEC*) val=12 ;;
      esac

      cal $val $year
}
```

OUTPUT:

```
surbhi@ubuntu:~/Desktop/SPprac/programs$ ./Q11_2.sh jan mar nov
    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
```

```
     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

   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
```

12. Write a Shell program to modify the inbuilt cal program to be able to handle following input:

$ cal jan….oct

```
argc=$#

if [ $argc -gt 1 ]
then
echo -e "\nOnly one argument should be given.\n";
exit
fi

arg=$1
arg1=`echo $arg | tr -s "."`
months=`echo $arg1 | tr "." " "`

month1=`echo $months | cut -d " " -f1`
month2=`echo $months | cut -d " " -f2`

#year=`date | cut -d " " -f6`
set `date`
   year=$6
   #echo $month1
   #echo $month2

case $month1 in
Jan*|JAN*) val1=1;;
Feb*|FEB*) val1=2;;
Mar*|MAR*) val1=3 ;;
Apr*|APR*) val1=4;;
May|MAY) val1=5 ;;
Jun*|JUN*) val1=6 ;;
Jul*|JUL*) val1=7 ;;
Aug*|AUG*) val1=8 ;;
Sep*|SEP*) val1=9;;
Oct*|OCT*) val1=10 ;;
Nov*|NOV*) val1=11 ;;
Dec*|DEC*) val1=12 ;;
esac

case $month2 in
Jan*|JAN*) val2=1;;
Feb*|FEB*) val2=2;;
Mar*|MAR*) val2=3 ;;
Apr*|APR*) val2=4;;
May|MAY) val2=5 ;;
Jun*|JUN*) val2=6 ;;
Jul*|JUL*) val2=7 ;;
```

```
      Aug*|AUG*) val2=8 ;;
      Sep*|SEP*) val2=9;;
      Oct*|OCT*) val2=10 ;;
      Nov*|NOV*) val2=11 ;;
      Dec*|DEC*) val2=12 ;;
      esac

      #echo $val1
      #echo $val2

      if [ $val1 -gt  $val2 ]
      then
            echo -e "\nEntered range is invalid\n"
            exit
      fi

      for((i=val1;i<=val2;i++))
      do
      cal $i $year
      done
```

OUTPUT:

```
surbhi@ubuntu:~/Desktop/SPprac/programs$ ./Q12.sh feb...may
   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
                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
```