LAB INSTRUCTIONS

SHELL SCRIPTING

PURPOSES:

- To understand the basic structure and syntax of a UNIX/LINUX Bourne shell script
- To practice basic shell programming tools under the Bourne shell.

Copy the scripts below and execute them on the linux shell available to you. After exercising with the scripts, do the assignment that follows.

The following scripts have been written for you. Just copy into vi editor environments and execute.

1. *The simplest shell script – Echo command*
2. *Summation of two integers – If block*
3. *Summation of two real numbers – bc (basic calculator) command*
4. *Script to find out the biggest number in 3 numbers – If –elif block*
5. *Operation (summation, subtraction, multiplication and division) of two numbers – Switch*
6. *Script to reverse a given number – While block*
7. *A more complicated greeting shell script*
8. *Sort the given five numbers in ascending order (using array) – Do loop and array*
9. *Calculating average of given numbers on command line arguments – Do loop*
10. *Calculating factorial of a given number – While block*

1. UNIX shell and How to Change your Shell

   A UNIX shell is a kernel working environment. There are many UNIX shells in common use, such as `bash`, `ksh`, `csh`, and `tcsh`. Many scientific applications in AFS package space such as `GaussView`, `Cerius2`, `SYBYL`, `Cambridge Database`, etc., work only in the C shell environment, which is either `csh` or `tcsh`.

   To know which shell you are currently use, type at the prompt on any of the servers

   ```
   echo $SHELL
   ```

BOURNE SHELL SCRIPTS

1. The simplest, Hello script!

```
#!/bin/sh

echo "Hello, $LOGNAME!"
echo "Current date is `date`"
echo "User is `whoami`"
echo "Current directory `pwd`"
```

2. Summation of two integers

#!/bin/sh

```
if [ $# -ne 2 ]
then
   echo "Usage - $0   x    y"
   echo "Where x and y are two integers for which I will print sum"
   exit 1
fi
sum=`expr $1 + $2 `
echo "Sum of $1 and $2 is $sum "
```

3. Summation of two real numbers

```
#!/bin/sh

a=5.66
b=8.67
c=`echo $a + $b | bc`
echo "$a + $b = $c"
```

4. Script to find out the biggest number in 3 integers

```
#!/bin/sh

if [ $# -ne 3 ]
then
      echo "$0: number1 number2 number3 are not given" >&2
     exit 1
fi
n1=$1
n2=$2
n3=$3
if [ $n1 -gt $n2 ] && [ $n1 -gt $n3 ]
then
      echo "$n1 is Bigest number"
elif [ $n2 -gt $n1 ] && [ $n2 -gt $n3 ]
then
      echo "$n2 is Bigest number"
elif [ $n3 -gt $n1 ] && [ $n3 -gt $n2 ]
then
     echo "$n3 is Bigest number"
elif [ $1 -eq $2 ] && [ $1 -eq $3 ] && [ $2 -eq $3 ]
then
      echo "All the three numbers are equal"
else
      echo "I can not figure out which number is biger"
fi
```

5. Operation (summation, subtraction, multiplication and division) of two integers

```
#!/bin/sh

if test $# = 3
then
      case $2 in
       +) let z=$1+$3;;
       -) let z=$1-$3;;
       /) let z=$1/$3;;
       x|X) let z=$1*$3;;
       *) echo Warning - $2 invalid operator, only +,-,x,/ operator allowed
          exit;;
      esac
      echo Answer is $z
else
      echo "Usage - $0   value1  operator value2"
      echo "        Where, value1 and value2 are numeric values"
      echo "               operator can be +,-,/,x (For Multiplication)"
fi
```

6. Script to reverse a given positive integer

```
#!/bin/sh

if [ $# -ne 1 ]
then
    echo "Usage: $0   number"
    echo "       I will find reverse of given positive integer"
    echo "       For eg. $0 123, I will print 321"
    exit 1
fi

n=$1
rev=0; sd=0

while [ $n -gt 0 ]
do
    sd=`expr $n % 10`
    rev=`expr $rev \* 10  + $sd`
    n=`expr $n / 10`
done
echo  "Reverse number is $rev"
```

7. A more complicated greeting shell script

```
#!/bin/sh

temph=`date | cut -c12-13`
dat=`date +"%A %d in %B of %Y (%r)"`

if [ $temph -lt 12 ]
then
    mess="Good Morning $LOGNAME, Have nice day!"
fi

if [ $temph -gt 12 -a $temph -le 16 ]
then
    mess="Good Afternoon $LOGNAME"
fi

if [ $temph -gt 16 -a $temph -le 18 ]
then
    mess="Good Evening $LOGNAME"
fi
echo -e "$mess\nThis is $dat"
```

8. Sort the given five integer numbers in ascending order (using array)

```
#!/bin/sh

# Declare the array of 5 subscripts to hold 5 numbers
nos=(4 -1 2 66 10)

# Prints the number before sorting
echo "Original Numbers in array:"
for (( i = 0; i <= 4; i++ ))
do
  echo ${nos[$i]}
done

# Now do the Sorting of numbers
for (( i = 0; i <= 4 ; i++ ))
do
   for (( j = $i; j <= 4; j++ ))
   do
      if [ ${nos[$i]} -gt ${nos[$j]}  ]; then
           t=${nos[$i]}
           nos[$i]=${nos[$j]}
           nos[$j]=$t
      fi
   done
done

# Print the sorted number
echo -e "\nSorted Numbers in Ascending Order:"
for (( i=0; i <= 4; i++ ))
do
  echo ${nos[$i]}
done
```

9. Calculating average of given integer numbers on command line arguments

```
#!/bin/sh

avg=0
temp_total=0
number_of_args=$#

# First see the sufficent cmd args
if [ $# -lt 2 ] ; then
   echo -e "Oops! I need atleast 2 command line args\n"
   echo -e "Syntax: $0: number1 number2 ... numberN\n"
   echo -e "Example:$0 5 4\n\t$0 56 66 34"
   exit 1
fi

# now calculate the average of numbers given on command line as cmd args
for i in $*
do
   #  addition of all the numbers on cmd args
   temp_total=`expr $temp_total + $i `
```

```
done

avg=`expr $temp_total / $number_of_args `
echo "Average of all number is $avg"
```

10. Calculating factorial of a given integer number

```
#!/bin/sh

n=0; on=0
fact=1
echo -n "Enter number to find factorial : "
read n

on=$n
while [ $n -ge  1 ]
do
  fact=`expr $fact \* $n`
  n=`expr $n - 1`
done
echo "Factorial for $on is $fact"
```

==========

ASSIGNMENT I [DUE ON WEDNESDAY 4TH NOVEMBER 2015]

Write a shell script that checks whether the username and a password entered by a user are correct. The script should only allow a maximum of 3 attempts before locking a user out of the system

==========