LAB-2

1. Write a Shell Script which works like a calculator and performs below operations 
   Addition, Subtract, Division and Multiplication.

   ```bash
   echo "Enter number 1:" 
   read a 
   echo "Enter number 2:" 
   read b 

   echo "Operations:" 
   echo "1. Addition" 
   echo "2. Subtraction" 
   echo "3. Multiplication" 
   echo "4. Division" 
   echo "Enter Choice :"

   read ch 

   # For Addition
   if [ $ch == 1 ]
   then
       ans=$(( $a + $b ))
       echo "Addition = $ans"
   fi

   # For Substraction
   elif [ $ch == 2 ]
   then
       ans=$(( $a - $b ))
       echo "Substraction = $ans"
   fi

   # For Multiplication
   elif [ $ch == 3 ]
   then
       ans=$(( $a * $b ))
       echo "Multiplication = $ans"
   fi

   # For Division
   elif [ $ch == 4 ]
   then
       if [ $b != 0 ]
       then
           ans=$(( $a / $b ))
           echo "Division = $ans"
       else
           echo "Please Enter non-zero value in number 2..."
       fi
   else
       #Invalid Choice
   fi
   ```
Lab Solution – Shell Script

2. Write a Shell Script to find a greater number out of 2 numbers.

```bash
if [ $a -gt $b ]
then
    echo "a=$a is greater"
elif [ $b -gt $a ]
then
    echo "b=$b is greater"
else
    echo "a and b are equal"
fi
```

LAB-3

1. Write a Shell Script to check given number is ODD or EVEN.

```bash
if [ `expr $n % 2` -eq 0 ]
then
    echo "Number is even"
else
    echo "Number is odd"
fi
```

2. Write a Shell Script to find the largest among the 3 given numbers using if...else.

```bash
if [ $a -ge $b -a $a -ge $c ]
then
    echo "a is max"
fi

if [ $b -ge $a -a $b -ge $c ]
then
    echo "b is max"
fi
```
3. Write a Shell Script to find the largest among the 3 given numbers using nested if.
   echo "Enter first number: "
   read a
   echo "Enter second number: "
   read b
   echo "Enter third number: "
   read c
   # Using Nested if statement....
   if [ $a -eq $b ]
     then
     echo "All are equal"
   else
     if [ $a -ge $b ]
       then
       if [ $a -ge $c ]
         then
         echo "a is max"
       else
         echo "c is max"
       fi
     else
     if [ $b -ge $c ]
       then
       echo "b is max"
     else
       echo "c is max"
     fi
   fi
   fi
   fi

4. Write a Shell Script to find the largest among the 3 given numbers using if-else-if ladder.
   echo "Enter first number: "
   read a
   echo "Enter second number: "
   read b
   echo "Enter third number: "
   read c
   # Using if-else-if ladder
LAB-4

1. Write a Shell Script to check given year is Leap year or not.
   echo "Enter Year:"
   read y

   year=$y

   y=$(( y % 4 ))
   if [ $y -eq 0 ]
   then
     echo "$year is Leap Year!"
   else
     echo "$year is not a Leap Year!"
   fi

2. Write a shell script to generate mark sheet of a student. Take 3 subjects, calculate and display total marks, percentage and Class obtained by the student.
   echo "Enter Subject-1 marks: "
   read s1

   echo "Enter Subject-2 marks: "
   read s2

   echo "Enter Subject-3 marks: "
   read s3

   total=$(( s1 + s2 + s3 ))
   echo "Total Marks : $total"

   per=$(( total / 3 ))
   echo "Percentage: $per %"

   if [ $per -gt 70 ]
   then
     echo "Distiction Class"
   elif [ $per -ge 61 -a $per -le 70 ]
   then
     echo "First Class"
   elif [ $per -ge 51 -a $per -le 60 ]
   then
     echo "Second Class"
   else
     echo "Third Class"
   fi
then
  echo "Second Class"
elif [ $per -ge 41 -a $per -le 50 ]
  then
  echo "Third Class"
else
  echo "Fail"
fi

LAB-5

1. Write a Shell Script to print 1 to n numbers using while loop.
   echo "Enter n : "
   read n
   i=1
   echo "Number list:-"
   while [ $i -le $n ]
     do
       echo $i
       i=`expr $i + 1`
     done

2. Write a Shell Script to find Sum & Average of n numbers.
   echo "Enter Size(N)"
   read N
   i=1
   sum=0

   echo "Enter Numbers"
   while [ $i -le $N ]
     do
       read num               #get number
       sum=`expr $sum + $num`     #sum+=num
       i=`expr $i + 1`
     done
   avg=`expr $sum / $N`
   echo "Sum is $sum"
   echo "Average is $avg"

3. Write a Shell Script to display the multiplication table of the given number.
   echo "Enter a Number"
   read n
   i=1
   while [ $i -le 10 ]
     do
       echo "$n x $i = `expr $n \* $i`"
       i=`expr $i + 1`
     done

LAB-6
1. Write a Shell Script to find factorial of given number n.
   ```bash
   echo "Enter number"
   read num

   fact=1
   while [ $num -gt 1 ]
   do
     fact=$((fact * num))  #fact = fact * num
     num=$((num - 1))      #num = num - 1
   done

   echo $fact
   ```

2. Write a Shell Script to check whether a given number is palindrome or not.
   ```bash
   echo "Enter number"
   read number

   realnumber=$number
   n=`expr $number % 10`
   number=`expr $number / 10`
   while [[ $number -ne 0 ]]
   do
     n=`expr $n \* 10 + $number % 10`
     number=`expr $number / 10`
   done

   if [[ $n -eq $realnumber ]]; then
     echo "Palindrome"
   else
     echo "Not Palindrome"
   fi
   ```

LAB-7

1. Write a Shell Script to find the value of one number raised to the power of another.
   ```bash
   echo "Input number"
   read no
   echo "Input power"
   read power

   counter=0
   ans=1
   while [ $power -ne $counter ]
   do
     ans=`expr $ans \* $no`
     counter=`expr $counter + 1`
   done

   echo "$no power of $power is $ans"
   ```
2. Write a Shell Script to check whether a given number is prime or not.
   ```bash
   echo "Enter number"
   read number
   prime=1
   for (( i = 2; i < $number; i++ ))
   do
     if [[ `expr $number % $i` -eq 0 ]]
       then
         prime=0
     fi
   done
   if [[ $prime -eq 1 ]]
   then
     echo "Prime"
   else
     echo "Not Prime"
   fi
   ```

3. Write a Shell Script in which will accept a number n and display first m prime numbers as output?
   ```bash
   echo "enter start no="
   read i
   echo "enter end no="
   read j
   flag=0
   tem=2
   count=1
   while [ $i -ne $j ]
   do
     temp=`echo $i`
     while [ $temp -ne $tem ]
     do
       temp=`expr $temp - 1`
       n=`expr $i % $temp`
       if [ $n -eq 0 -a $flag -eq 0 ]
       then
         flag=1
       fi
     done
     if [ $flag -eq 0 ]
     then
       echo " prime No $count = $i"
       count=`expr $count + 1`
     else
     fi
     i=`expr $i + 1`
   ```
Lab Solution – Shell Script

LAB-8

1. Write a Shell Script to print the pyramid structure for the given number.
   ```bash
   echo "Enter size of pyramid :"
   read n
   i=0
   for((k=1; k<=n; k++))
   do
     # This loop print spaces
     for((a=i; a<=n; a++))
     do
       printf " ";
     done
     # This loop print the left side of the pyramid
     for((j=1; j<=k; j++))
     do
       printf "*";
     done
     # This loop print right side of the pyramid.
     for((i=1; i<k; i++))
     do
       printf "*";
     done
     # New line
     echo;
   done
   ```

2. Write a Shell Script to check whether the given string is palindrome or not.
   ```bash
   echo "Enter String"
   read String
   size=`expr length "$String``
   msg="Palindrome"
   for (( i = 0, j = size-1; i < size; i++,j-- )); do
     if [[ ${String:i:1} != ${String:j:1} ]]; then
       msg="Not Palindrome"
     fi
   done
   echo "String is $msg"
   ```

LAB-9

1. Write a Shell Script to find first n Fibonacci numbers like: 0 1, 1, 2, 3, 5, 13
   ```bash
   echo "Enter Range"
   read number
   ```
n1=1
n2=1
echo "Fibonacci Series:-"
for (( i = 0; $i < $number; i++ )); do
    echo $n2
temp=$n1
    n1=`expr $n1 + $n2`
    n2=$temp
done

2. To scan the name of the command and executes it.
echo "Command:-"
echo "1.cal"
echo "2.ls"
echo "3.date"
echo "Enter choice:" read choice
case $choice in
  1)
    echo $(cal)
    ;;
  2)
    echo $(ls)
    ;;
  3)
    echo $(date)
    ;;
esac

3. Write a Shell Script to display current month calendar.
echo "Current Month Calendar"
echo $(cal)

LAB-10
1. Write a Shell Script which will print the following menu and execute the given task?
a) Display calendar of current month
b) Display today's date and time
c) Display usernames that are currently logged in the system
d) Display your name at given x, y position
e) Display your terminal number
echo "1. Display calendar of current month"
echo "2. Display today's date and time"
echo "3. Display usernames that are currently logged in the system"
echo "4. Display your name at given x, y position"
echo "5. Display your terminal number"
echo "Enter choice:" read choice
case $choice in
1) echo $(cal) ;;
2) echo $(date) ;;
3) echo $(whoami) ;;
4) printf "%20s%30s
" Maulik Trivedi ;;
5) echo $(tty) ;;
esac

LAB-11

1. Write a Shell Script to read n numbers as command arguments and sort them in descending order.
   
echo "Enter N"
read n
for (( i = 1; i <= n; i++ )); do
   echo "Enter Array ["$i"] : "
   read array[i]
done
for (( i = 1; i <= 5; i++ )); do
   for (( j = i+1; j <= 5; j++ )); do
      if [[ ${array[i]} > ${array[j]} ]]; then
         temp=${array[i]}
         array[i]=${array[j]}
         array[j]=$temp
      fi
   done
done
echo "Elements of sorted array:-"
for (( i = 1; i <= 5; i++ )); do
   echo ${array[i]}
done

2. Write a Shell Script to display all executable files, directories and zero sized files from current directory.
   #display empty file
   find /home/student/Documents/ -type f -empty
   
   #display directory
   $ ls -d */
   
   #display executable file
   find /home/student/.anaconda/navigator/ -executabel -type f
3. Write a Shell Script to display the date, time and a welcome message (like Good Morning etc.). The time should be displayed with a.m. or p.m. and not in 24 hours notation.

```bash
HH=`date +%H`
time=`date +%M %p`
if [ $HH -ge 12 ]; then
    HH=`expr $HH % 12`
    if [ $HH -lt 5 ]; then
        msg="GOOD AFTERNOON"
    elif [ $HH -ge 5 ] && [ $HH -lt 9 ]; then
        msg="GOOD EVENING"
    else
        msg="GOOD NIGHT"
    fi
    echo "$msg , CURRENT TIME $HH:$time" exit 1
else
    if [ $HH -lt 5 ]; then
        msg="GOOD NIGHT"
    else
        msg="GOOD MORNING"
    fi
    echo "$msg , CURRENT TIME $HH:$time"
fi
```

LAB-12

1. Write a Shell Script to validate the entered date. (E.g. Date format is: dd-mm-yyyy).

```bash
# store day, month and year
dd=0
mm=0
yy=0

# store number of days in a month
days=0

# get day, month and year
echo -n "Enter day (dd) : "
read dd

echo -n "Enter month (mm) : "
read mm

echo -n "Enter year (yyyy) : "
read yy

# if month is negative (<0) or greater than 12
# then it is invalid month
```
if [ $mm -le 0 -o $mm -gt 12 ];
then
    echo "$mm is invalid month."
    exit 1
fi

# Find out number of days in given month
case $mm in
    1) days=31;;
    2) days=28 ;;
    3) days=31 ;;
    4) days=30 ;;
    5) days=31 ;;
    6) days=30 ;;
    7) days=31 ;;
    8) days=31 ;;
    9) days=30 ;;
   10) days=31 ;;
   11) days=30 ;;
   12) days=31 ;;
   *) days=-1;;
esac

# find out if it is a leap year or not
if [ $mm -eq 2 ]; # if it is feb month then only check of leap year
then
    if [ $((yy % 4)) -ne 0 ]; then
        # not a leap year : means do nothing and use old value of days
    elif [ $((yy % 400)) -eq 0 ]; then
        # yes, it's a leap year
        days=29
    elif [ $((yy % 100)) -eq 0 ]; then
        # not a leap year do nothing and use old value of days
    else
        # it is a leap year
        days=29
    fi
fi

# if day is negative (<0) and if day is more than
# that months days then day is invalid
if [ $dd -le 0 -o $dd -gt $days ];
then
    echo "$dd day is invalid"
    exit 3
fi

# if no error that means date dd/mm/yyyy is valid one
echo "$dd/$mm/$yy is a valid date"

2. Write a Shell Script which checks whether a given user is valid or not.
   echo “enter user :”
   read user
3. Write a Shell Script that finds total no. of users and finds out how many of them are currently logged in.

```bash
cat /etc/passwd>user.txt
set `wc -l user.txt`
log=`who -a | wc -l`
echo "There are $1 users"
echo "There are $log user logged in currently "
```

4. Write an awk program using function, which convert each word in a given text into capital.

```bash
a="UPPER CASE"
echo "$a" | awk '{print tolower($0)}'
```

5. Write a program for process creation using C. (Use of gcc compiler).

```c
#include <stdio.h>
#include <sys/types.h>
#include <unistd.h>

int main()
{
    // make two process which run same program after this instruction
    fork();

    printf("Hello world!\n");
    return 0;
}
```

Output:
```
Hello world!
Hello world!
```