

# OPERATING SYSTEM

## (4ITRC2)

### LAB ASSIGNMENT 3

**Aim:** To create shell scripts for the following questions.

**To perform:** To code and solve the following problem.

**To Submit:**

1.To find largest of three Numbers.

```
echo "Enter three numbers:"
read a b c
if [ $a -gt $b ] && [ $a -gt $c ]; then
echo "Largest number is $a"
elif [ $b -gt $a ] && [ $b -gt $c ]; then
echo "Largest number is $b"
else
echo "Largest number is $c"
fi
```

2.To find a year is leap year or not.

```
echo "Enter a year:"
read year
if (( (year % 4 == 0 && year % 100 != 0) || (year % 400 == 0) )); then
echo "$year is a leap year."
else
echo "$year is not a leap year."
fi
```

3.To input angles of a triangle and find out whether it is valid triangle or not.

```
echo "Enter three angles:"
read x y z
```

```

sum=$((x + y + z))
if [ $sum -eq 180 ]; then
    echo "Valid Triangle"
else
    echo "Invalid Triangle"
fi

```

4.To check whether a character is alphabet, digit or special character.

```

echo "Enter a character:"
read char
if [[ "$char" =~ [a-zA-Z] ]]; then
    echo "Alphabet"
elif [[ "$char" =~ [0-9] ]]; then
    echo "Digit"
else
    echo "Special Character"
fi

```

5.To Calculate profit or loss.

```

echo "Enter Cost Price and Selling Price:"
read cp sp
diff=$((sp - cp))
if [ $diff -gt 0 ]; then
    echo "Profit: $diff"
elif [ $diff -lt 0 ]; then
    echo "Loss: ${diff#-}"
else
    echo "No Profit No Loss"
fi

```

6.To print all even and odd numbers from 1 to 10.

```

echo "Even numbers:"
for ((i=2; i<=10; i+=2)); do echo $i; done
echo "Odd numbers:"
for ((i=1; i<=10; i+=2)); do echo $i; done

```

**7.To Print table of a given number.**

```
echo "Enter a number:"
read num
for ((i=1; i<=10; i++)); do
    echo "$num x $i = $((num * i))"
done
```

**8.To Find factorial of a number.**

```
echo "Enter a number:"
read n
fact=1
for ((i=1; i<=n; i++)); do
    fact=$((fact * i))
done
echo "Factorial of $n is $fact"
```

**9.To Print sum of all even numbers from 1 to 10.**

```
sum=0
for ((i=2; i<=10; i+=2)); do
    sum=$((sum + i))
done
echo "Sum of even numbers from 1 to 10 is $sum"
```

**10.To Print sum of digits of any number.**

```
echo "Enter a number:"
read num
sum=0
while [ $num -gt 0 ]; do
    digit=$((num % 10))
    sum=$((sum + digit))
    num=$((num / 10))
done
echo "Sum of digits is $sum"
```

**11.**To make a basic calculator which performs addition, subtraction, Multiplication, division.

```
echo "Enter two numbers:"
read a b
echo "Enter operation (+ - * /):"
read op
case $op in
+) echo "Result: $((a + b))" ;;
-) echo "Result: $((a - b))" ;;
\*) echo "Result: $((a * b))" ;;
/) echo "Result: $((a / b))" ;;
*) echo "Invalid operation" ;;
esac
```

**12.**To Print days of a week.

```
echo "Days of the week:"
echo -e
"Sunday\nMonday\nTuesday\nWednesday\nThursday\nFriday\nSaturday"
```

**13.**To Print first 4 months with 31 days.

```
echo "January\nMarch\nMay\nJuly"
```

**14.**Using functions.

**(a).**Check if a number is an Armstrong number.

```
is_armstrong() {
num=$1
sum=0
temp=$num
while [ $temp -gt 0 ]; do
digit=$((temp % 10))
sum=$((sum + digit**3))
temp=$((temp / 10))
done
if [ $sum -eq $num ]; then
echo "$num is an Armstrong number."
else
```

```

    echo "$num is not an Armstrong number."
fi
}

```

**(b).** Check if a number is a palindrome.

```

is_palindrome() {
    num=$1
    rev=$(echo $num | rev)
    if [ "$num" -eq "$rev" ]; then
        echo "$num is a palindrome."
    else
        echo "$num is not a palindrome."
    fi
}

```

**(c).** Print Fibonacci series up to n terms.

```

fibonacci() {
    n=$1
    a=0
    b=1
    echo -n "$a $b "
    for ((i=2; i<=n; i++)); do
        c=$((a + b))
        echo -n "$c "
        a=$b
        b=$c
    done
    echo
}

```

**(d).** Check if a number is prime or composite.

```

is_prime() {
    num=$1
    if [ $num -lt 2 ]; then echo "Not prime"; return; fi
    for ((i=2; i*i<=num; i++)); do
        if [ $(num % i) -eq 0 ]; then
            echo "$num is composite"

```

```
        return
    fi
done
echo "$num is prime"
}
```

**(e).**Convert decimal to binary.

```
dec_to_bin() {
    num=$1
    echo "Binary equivalent: $(echo "obase=2; $num" | bc)"
}
```

Milind Sharma2414193