**4ITRC2 Operating System Lab**

**Lab Assignment 3**

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

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

**To Submit: Give shell scripts for following:**

**1. To find Largest of Three Numbers**

echo "Enter three numbers:"

read a b c

if [ $a -ge $b ] && [ $a -ge $c ]; then

echo "$a is the largest"

elif [ $b -ge $a ] && [ $b -ge $c ]; then

echo "$b is the largest"

else

echo "$c is the largest"

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 of a triangle:"

read a b c

sum=$((a + b + c))

if [ $sum -eq 180 ]; then

echo "It is a valid triangle"

else

echo "It is not a valid triangle"

fi

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

echo "Enter a character:"

read ch

if [[ $ch =~ [a-zA-Z] ]]; then

echo "Alphabet"

elif [[ $ch =~ [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

if [ $sp -gt $cp ]; then

profit=$((sp - cp))

echo "Profit: $profit"

elif [ $cp -gt $sp ]; then

loss=$((cp - sp))

echo "Loss: $loss"

else

echo "No Profit No Loss"

fi

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

echo "Even Numbers:"

for i in {1..10}; do

if [ $((i % 2)) -eq 0 ]; then echo $i; fi

done

echo "Odd Numbers:"

for i in {1..10}; do

if [ $((i % 2)) -ne 0 ]; then echo $i; fi

done

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

echo "Enter a number:"

read num

for i in {1..10}; do

echo "$num \* $i = $((num \* i))"

done

**8. To find factorial of a given integer**

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 in {1..10}; do

if [ $((i % 2)) -eq 0 ]; then

sum=$((sum + i))

fi

done

echo "Sum of even numbers: $sum"

**10. To print sum of digit of any number.**

echo "Enter a number:"

read n

sum=0

while [ $n -gt 0 ]; do

digit=$((n % 10))

sum=$((sum + digit))

n=$((n / 10))

done

echo "Sum of digits: $sum"

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

**division**

echo "Enter two numbers:"

read a b

echo "Choose operation: + - \* /"

read op

case $op in

+) echo "$((a + b))";;

-) echo "$((a - b))";;

\\*) echo "$((a \* b))";;

/) echo "$((a / b))";;

\*) echo "Invalid operator";;

esac

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

for day in Monday Tuesday Wednesday Thursday Friday Saturday Sunday; do

echo $day

done

**13. To print starting 4 months having 31 days.**

months=(January March May July August October December)

for ((i=0; i<4; i++)); do

echo ${months[$i]}

done

**14. Using functions,**

**a. To find given number is Amstrong number or not**

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

}

echo "Enter a number:"

read n

is\_armstrong $n

**b. To find whether a number is palindrome or not**

is\_palindrome() {

n=$1

rev=0

temp=$n

while [ $temp -gt 0 ]; do

digit=$((temp % 10))

rev=$((rev \* 10 + digit))

temp=$((temp / 10))

done

if [ $n -eq $rev ]; then

echo "$n is a palindrome"

else

echo "$n is not a palindrome"

fi

}

echo "Enter a number:"

read n

is\_palindrome $n

**c. To print Fibonacci series upto n terms**

fibonacci() {

a=0

b=1

echo "Fibonacci Series:"

for (( i=0; i<$1; i++ )); do

echo -n "$a "

fn=$((a + b))

a=$b

b=$fn

done

echo

}

echo "Enter number of terms:"

read n

fibonacci $n

**d. To find given number is prime or composite**

is\_prime() {

n=$1

if [ $n -lt 2 ]; then echo "Not Prime"; return; fi

for ((i=2; i\*i<=n; i++)); do

if [ $((n % i)) -eq 0 ]; then

echo "$n is Composite"

return

fi

done

echo "$n is Prime"

}

echo "Enter a number:"

read n

is\_prime $n

**e. To convert a given decimal number to binary equivalent**

echo "Enter a decimal number:"

read dec

echo "Binary: $(echo "obase=2;$dec" | bc)"