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

read -p "Enter three numbers: " a b c

if (( a >= b && a >= c )); then

echo "$a is largest"

elif (( b >= a && b >= c )); then

echo "$b is largest"

else

echo "$c is largest"

fi

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

read -p "Enter year: " y

if (( y % 400 == 0 || (y % 4 == 0 && y % 100 != 0) )); then

echo "$y is a leap year"

else

echo "$y is not a leap year"

fi

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

read -p "Enter three angles: " a b c

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

if (( sum == 180 && a > 0 && b > 0 && c > 0 )); then

echo "Valid Triangle"

else

echo "Invalid Triangle"

fi

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

read -p "Enter a character: " ch

if [[ $ch =~ [A-Za-z] ]]; then

echo "Alphabet"

elif [[ $ch =~ [0-9] ]]; then

echo "Digit"

else

echo "Special Character"

fi

1. To calculate profit or loss

read -p "Enter Cost Price: " cp

read -p "Enter Selling Price: " sp

if (( sp > cp )); then

echo "Profit: $((sp - cp))"

elif (( cp > sp )); then

echo "Loss: $((cp - sp))"

else

echo "No Profit No Loss"

fi

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

echo "Even numbers:"

for ((i=1; i<=10; i++)); do

if (( i % 2 == 0 )); then echo $i; fi

done

echo "Odd numbers:"

for ((i=1; i<=10; i++)); do

if (( i % 2 != 0 )); then echo $i; fi

done

1. To print table of a given number

read -p "Enter a number: " num

for ((i=1; i<=10; i++)); do

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

done

1. To find factorial of a given integer

read -p "Enter a number: " n

fact=1

for ((i=1; i<=n; i++)); do

fact=$((fact \* i))

done

echo "Factorial: $fact"

1. 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: $sum"

1. To print sum of digit of any number.

read -p "Enter a number: " num

sum=0

while (( num > 0 )); do

sum=$((sum + num % 10))

num=$((num / 10))

done

echo "Sum of digits: $sum"

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

read -p "Enter first number: " a

read -p "Enter second number: " b

read -p "Choose operation (+ - \\* /): " op

case $op in

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

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

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

/)

if (( b == 0 )); then

echo "Division by zero error"

else

echo "$((a / b))"

fi ;;

\*) echo "Invalid operation" ;;

esac

1. To print days of a week.

days=("Sunday" "Monday" "Tuesday" "Wednesday" "Thursday" "Friday" "Saturday")

for day in "${days[@]}"; do

echo "$day"

done

1. To print starting 4 months having 31 days.

months=("January" "March" "May" "July")

for month in "${months[@]}"; do

echo "$month"

done

1. Using functions,
   1. To find given number is Amstrong number or not

armstrong() {

num=$1 sum=0 n=$num

while (( n > 0 )); do

digit=$((n % 10))

sum=$((sum + digit\*\*3))

n=$((n / 10))

done

[[ $sum -eq $num ]] && echo "Armstrong" || echo "Not Armstrong"

}

read -p "Enter number: " num

armstrong $num

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

palindrome() {

n=$1 rev=0 orig=$n

while (( n > 0 )); do

rev=$((rev \* 10 + n % 10))

n=$((n / 10))

done

[[ $rev -eq $orig ]] && echo "Palindrome" || echo "Not Palindrome"

}

read -p "Enter number: " num

palindrome $num

* 1. To print Fibonacci series upto n terms

fibonacci() {

n=$1

a=0 b=1

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

echo -n "$a "

fn=$((a + b))

a=$b

b=$fn

done

echo

}

read -p "Enter number of terms: " num

fibonacci $num

* 1. To find given number is prime or composite

prime\_check() {

n=$1

if (( n <= 1 )); then

echo "Neither prime nor composite"

return

fi

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

if (( n % i == 0 )); then

echo "Composite"

return

fi

done

echo "Prime"

}

read -p "Enter number: " num

prime\_check $num

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

decimal\_to\_binary() {

n=$1 bin=""

while (( n > 0 )); do

bin=$((n % 2))$bin

n=$((n / 2))

done

echo "Binary: $bin"

}

read -p "Enter decimal number: " num

decimal\_to\_binary $num