**4ITRC2 Operating System Lab**

**Lab Assignment 3**

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

echo "$year is a leap year"

else

echo "$year is not a leap year"

fi

**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 ] && [ $a -gt 0 ] && [ $b -gt 0 ] && [ $c -gt 0 ]; 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

case $char in

[a-zA-Z]) echo "Alphabet" ;;

[0-9]) echo "Digit" ;;

\*) echo "Special character" ;;

esac

**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 == 0 )); then echo $i; fi

done

echo "Odd numbers:"

for i in {1..10}; do

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

done

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

echo "Enter a number:"

read n

for i in {1..10}; do

echo "$n x $i = $((n \* 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 == 0 )); then

sum=$((sum + i))

fi

done

echo "Sum of even numbers from 1 to 10 is $sum"

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

echo "Enter a number:"

read num

sum=0

while [ $num -ne 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 "Choose operation (+, -, \*, /):"

read op

case $op in

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

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

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

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

\*) echo "Invalid operator" ;;

esac

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

days=(Monday Tuesday Wednesday Thursday Friday Saturday Sunday)

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

echo "$day"

done

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

bmonths=(January March May July August October December)

echo "First 4 months having 31 days:"

for i in {0..3}; do

echo "${months[$i]}"

done

**14. Using functions,**

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

amstrong() {

read -p "Enter number: " n

temp=$n

sum=0

while [ $n -gt 0 ]; do

digit=$((n % 10))

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

n=$((n / 10))

done

if [ $sum -eq $temp ]; then

echo "Armstrong number"

else

echo "Not an Armstrong number"

fi

}

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

palindrome() {

read -p "Enter number: " n

temp=$n

rev=0

while [ $n -gt 0 ]; do

digit=$((n % 10))

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

n=$((n / 10))

done

if [ $rev -eq $temp ]; then

echo "Palindrome number"

else

echo "Not a palindrome"

fi

}

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

fibonacci() {

read -p "Enter terms: " n

a=0

b=1

echo "Fibonacci series:"

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

echo -n "$a "

fn=$((a + b))

a=$b

b=$fn

done

echo

}

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

prime() {

read -p "Enter number: " n

if [ $n -lt 2 ]; then echo "Not prime"

else

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

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

echo "Composite"

return

fi

done

echo "Prime number"

fi

}

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

decimal\_to\_binary() {

read -p "Enter decimal number: " n

binary=""

while [ $n -gt 0 ]; do

rem=$((n % 2))

binary="$rem$binary"

n=$((n / 2))

done

echo "Binary: $binary"

}