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 % 400 == 0 || (year % 4 == 0 && year % 100 != 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 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
```

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
if (( sp > cp )); then
  echo "Profit: $((sp - cp))"
elif (( cp > sp )); then
  echo "Loss: $((cp - sp))"
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 -n "$i "
  fi
done
echo -e "\nOdd numbers:"
for i in {1..10}; do
  if (( i % 2 != 0 )); then
    echo -n "$i "
  fi
done
echo
```

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

```
echo "Enter a number:"
read n
for ((i=1; i<=10; i++)); 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=2; i<=10; i+=2)); do
  sum=$((sum + i))
done
echo "Sum of even numbers from 1 to 10: $sum"
```

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

```
echo "Enter a number:"
read n
sum=0
while ((n != 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 "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=(Sunday Monday Tuesday Wednesday Thursday Friday Saturday)
for day in "${days[@]}"; do
  echo "$day"
done
```

13.To print starting 4 months having 31 days.

```
months=("January" "March" "May" "July" "August" "October"
"December")
echo "First four months with 31 days:"
for ((i=0; i<4; i++)); do
  echo "${months[i]}"
done
```

14. Using functions,

a. To find given number is Amstrong number or not

b. To find whether a number is palindrome or not

c. To print Fibonacci series upto n terms

d. To find given number is prime or composite

e. To convert a given decimal number to binary equivalent

```
amstrong() {
  echo "Enter number:"
  read num
  n=$num
```

```
  sum=0
  while ((n != 0)); do
    d=$((n % 10))
    sum=$((sum + d * d * d))
    n=$((n / 10))
  done
  [[ $sum -eq $num ]] && echo "Armstrong number" || echo "Not Armstrong"
}

palindrome() {
  echo "Enter number:"
  read n
  rev=0
  num=$n
  while ((n != 0)); do
    d=$((n % 10))
    rev=$((rev * 10 + d))
    n=$((n / 10))
  done
  [[ $rev -eq $num ]] && echo "Palindrome" || echo "Not Palindrome"
}

fibonacci() {
  echo "Enter number of terms:"
  read n
  a=0
```

```
  b=1
  echo -n "$a $b "
  for ((i=3; i<=n; i++)); do
    c=$((a + b))
    echo -n "$c "
    a=$b
    b=$c
  done
  echo
}

prime_check() {
  echo "Enter number:"
  read n
  if ((n <= 1)); then
    echo "Not prime"
    return
  fi
  for ((i=2; i*i<=n; i++)); do
    if ((n % i == 0)); then
      echo "Composite"
      return
    fi
  done
  echo "Prime"
}
```

```
decimal_to_binary() {
  echo "Enter a decimal number:"
  read n
  echo -n "Binary: "
  echo "obase=2; $n" | bc
}
```