Assignment 3

To code and give shell scripts for the following

1. To find Largest of Three Numbers

```
find_largest_of_three() {
   echo "Enter three numbers:"
   read num1 num2 num3
   if [ $num1 -ge $num2 ] && [ $num1 -ge $num3 ]; then
      largest=$num1
   elif [ $num2 -ge $num1 ] && [ $num2 -ge $num3 ]; then
      largest=$num2
   else
      largest=$num3
   fi
   echo "The largest number is: $largest"
}
```

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

```
check_leap_year() {
   echo "Enter a year:"
   read year
   if [ $(($year % 400)) -eq 0 ] || [ $(($year % 4)) -eq 0 -a $(($year % 100)) -ne 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

```
check_valid_triangle() {
   echo "Enter three angles of a triangle:"
   read angle1 angle2 angle3
   sum=$(($angle1 + $angle2 + $angle3))
   if [ $sum -eq 180 ] && [ $angle1 -gt 0 ] && [ $angle2 -gt 0 ] && [ $angle3 -gt 0 ]; then
      echo "This is a valid triangle"
   else
      echo "This is not a valid triangle"
   fi
}
```

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

```
check_character_type() {
   echo "Enter a character:"
   read char
   if [[ $char =~ [[:alpha:]] ]]; then
      echo "$char is an alphabet"
   elif [[ $char =~ [[:digit:]] ]]; then
      echo "$char is a digit"
   else
      echo "$char is a special character"
   fi
}
```

5. To calculate profit or loss

```
calculate_profit_loss() {
   echo "Enter cost price:"
   read cost_price
   echo "Enter selling price:"
   read selling_price

   if [ $selling_price -gt $cost_price ]; then
      profit=$(($selling_price - $cost_price))
      echo "Profit of $profit"
   elif [ $cost_price -gt $selling_price ]; then
      loss=$(($cost_price - $selling_price))
      echo "Loss of $loss"
   else
      echo "No profit, no loss"
   fi
}
```

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

```
print_even_odd() {
   echo "Even numbers from 1 to 10:"
   for (( i=1; i<=10; i++ )); do
      if [ $(($i % 2)) -eq 0 ]; then
         echo -n "$i "
      fi
   done
   echo
   echo "Odd numbers from 1 to 10:"
   for (( i=1; i<=10; i++ )); do
```

```
    if [ $(($i % 2)) -ne 0 ]; then
        echo -n "$i "
    fi
  done
  echo
}
```

7. To print table of a given number

```
print_table() {
  echo "Enter a number to print its table:"
  read num
  echo "Table of $num:"
  for (( i=1; i<=10; i++ )); do
    result=$(($num * $i))
    echo "$num × $i = $result"
  done
}
```

8. To find factorial of a given integer

```
find_factorial() {
  echo "Enter a number to find its factorial:"
  read num
  factorial=1
  for (( i=1; i<=num; i++ )); do
    factorial=$(($factorial * $i))
  done
  echo "Factorial of $num is $factorial"
}
```

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

```
sum_of_even_numbers() {
   sum=0
   for (( i=1; i<=10; i++ )); do
      if [ $(($i % 2)) -eq 0 ]; then
         sum=$(($sum + $i))
      fi
   done
   echo "Sum of all even numbers from 1 to 10 is $sum"
}
```

10. To print sum of digit of any number.

```
sum_of_digits() {
   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

```
basic_calculator() {
```

```
echo "Basic Calculator"
echo "Enter first number:"
read num1
echo "Enter second number:"
read num2
echo "Enter operation (+, -, *, /):"
read operation
case $operation in
  "+")
    result=$(($num1 + $num2))
    echo "Result: $result"
    ;;
  "-")
    result=$(($num1 - $num2))
    echo "Result: $result"
    ;;
  "*")
    result=$(($num1 * $num2))
    echo "Result: $result"
    ;;
  "/")
    if [ $num2 -eq 0 ]; then
      echo "Error: Division by zero"
    else
      result=$(echo "scale=2; $num1 / $num2" | bc)
      echo "Result: $result"
    fi
    ;;
```

```
    *)
        echo "Invalid operation"
        ;;
  esac
}
```

12. To print days of a week.

```
print_days_of_week() {
  echo "Days of the week:"
  echo "1. Monday"
  echo "2. Tuesday"
  echo "3. Wednesday"
  echo "4. Thursday"
  echo "5. Friday"
  echo "6. Saturday"
  echo "7. Sunday"
}
```

13. To print starting 4 months having 31 days.

```
print_months_31_days() {
  echo "First 4 months with 31 days:"
  echo "1. January (31 days)"
  echo "3. March (31 days)"
  echo "5. May (31 days)"
  echo "7. July (31 days)"
}
```

14. Using functions,

a. To find given number is Amstrong number or not

```
is_armstrong() {
        echo "Enter a number to check if it's an Armstrong number:"
```

```
    read num
    original=$num
    digits=${#num}
    sum=0

    while [ $num -gt 0 ]; do
      digit=$(($num % 10))
      power=1
      for (( i=0; i<digits; i++ )); do
        power=$(($power * $digit))
      done
      sum=$(($sum + $power))
      num=$(($num / 10))
    done

    if [ $sum -eq $original ]; then
      echo "$original is an Armstrong number"
    else
      echo "$original is not an Armstrong number"
    fi
  }
```

b. To find whether a number is palindrome or not

```
is_palindrome() {
  echo "Enter a number to check if it's a palindrome:"
  read num
  original=$num
  reverse=0
  while [ $num -gt 0 ]; do
```

```
    digit=$(($num % 10))
    reverse=$(($reverse * 10 + $digit))
    num=$(($num / 10))
  done
  if [ $original -eq $reverse ]; then
    echo "$original is a palindrome"
  else
    echo "$original is not a palindrome"
  fi
}
```

c. To print Fibonacci series upto n terms

```
print_fibonacci() {
  echo "Enter the number of terms for Fibonacci series:"
  read n
  a=0
  b=1
  echo "Fibonacci series up to $n terms:"
  echo -n "$a "
  if [ $n -gt 1 ]; then
    echo -n "$b "
  fi
  for (( i=3; i<=n; i++ )); do
    c=$(($a + $b))
    echo -n "$c "
    a=$b
    b=$c
  done
  echo
```

```
}
```

d. To find given number is prime or composite

```
is_prime() {
   echo "Enter a number to check if it's prime or composite:"
   read num

   if [ $num -lt 2 ]; then
      echo "$num is neither prime nor composite"
      return
   fi

   is_prime=1
   for (( i=2; i*i<=num; i++ )); do
      if [ $(($num % $i)) -eq 0 ]; then
         is_prime=0
         break
      fi
   done

   if [ $is_prime -eq 1 ]; then
      echo "$num is a prime number"
   else
      echo "$num is a composite number"
   fi
}
```

e. To convert a given decimal number to binary equivalent

```
decimal_to_binary() {
```

```
    echo "Enter a decimal number to convert to binary:"
    read decimal

    binary=""
    num=$decimal

    while [ $num -gt 0 ]; do
        remainder=$(($num % 2))
        binary="$remainder$binary"
        num=$(($num / 2))
    done

    echo "Binary equivalent of $decimal is $binary"
}
```