Name - Shubham Garg
Reg No – BL.EN.U4CSE20158

Lab 1

Question 1:

Write a program in Scala to implement a function to find the largest of 3 numbers.

Code:

```scala
object q1 {
    def findLargest(a: Int, b: Int, c: Int): Int = {
        if (a >= b && a >= c) a
        else if (b >= a && b >= c) b
        else c
    }

    def main(args: Array[String]) = {
        println("Largest of 3 numbers is: "+ findLargest(3, 6, 9))
    }
}
```

Output:

q1.scala

```scala
object q1 {
    def findLargest(a: Int, b: Int, c: Int): Int = {
        if (a ≥ b && a ≥ c) a
        else if (b ≥ a && b ≥ c) b
        else c
    }

    def main(args: Array[String]) = {
        println("Largest of 3 numbers is: "+ findLargest(3, 6, 9))
    }
}
```

PROBLEMS OUTPUT TERMINAL COMMENTS DEBUG CONSOLE

```
PS D:\Sem 6\PPL\Labs\Lab1> cd "d:\Sem 6\PPL\Labs\Lab1\" ; if ($?) { scalac q1.scala } ; if ($?) { scala q1 }
Largest of 3 numbers is: 9
PS D:\Sem 6\PPL\Labs\Lab1>
```

Name - Shubham Garg
Reg No – BL.EN.U4CSE20158

Question 2:

Write a program in Scala to check whether a numbered entered of user's choice is palindrome or not.

Code:

```
object q2 {
  def main(args: Array[String]): Unit = {
    print("Enter a number: ")
    val num = scala.io.StdIn.readInt()
    val numberString = num.toString
    val reversedString = numberString.reverse

    if (numberString == reversedString) {
      println(s"$num is a palindrome")
    } else {
      println(s"$num is not a palindrome")
    }
  }
}
```

Output:

q2.scala

```
object q2 {
  def main(args: Array[String]): Unit = {
    print("Enter a number: ")
    val num = scala.io.StdIn.readInt()
    val numberString = num.toString
    val reversedString = numberString.reverse

    if (numberString == reversedString) {
      println(s"$num is a palindrome")
    } else {
      println(s"$num is not a palindrome")
    }
  }
}
```

PROBLEMS OUTPUT TERMINAL COMMENTS DEBUG CONSOLE

```
PS D:\Sem 6\PPL\Labs\Lab1> cd "d:\Sem 6\PPL\Labs\Lab1\" ; if ($?) { scalac q2.scala } ; if ($?) { scala q2 }
Enter a number: 15951
15951 is a palindrome
PS D:\Sem 6\PPL\Labs\Lab1> cd "d:\Sem 6\PPL\Labs\Lab1\" ; if ($?) { scalac q2.scala } ; if ($?) { scala q2 }
Enter a number: 1234
1234 is not a palindrome
PS D:\Sem 6\PPL\Labs\Lab1>
```

Name - Shubham Garg
Reg No – BL.EN.U4CSE20158

Question 3:

Write a program in Scala to find the average of the array.

Code:

```
object q3 {
  def main(args: Array[String]): Unit = {
    val arr = Array(1, 2, 3, 4, 5, 6, 7, 8, 9)
    val avg = arr.sum.toFloat / arr.length.toFloat
    println("Average of the array is: "+avg)
  }
}
```

Output:

q3.scala

```
object q3 {
  def main(args: Array[String]): Unit = {
    val arr = Array(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)
    val avg = arr.sum.toFloat / arr.length.toFloat
    println("Average of the array is: "+avg)
  }
}
```

PROBLEMS OUTPUT TERMINAL COMMENTS DEBUG CONSOLE Code

```
PS D:\Sem 6\PPL\Labs\Lab1> cd "d:\Sem 6\PPL\Labs\Lab1\" ; if ($?) { scalac q3.scala } ; if ($?) { scala q3 }
Average of the array is: 5.5
PS D:\Sem 6\PPL\Labs\Lab1>
```

Name - Shubham Garg
Reg No – BL.EN.U4CSE20158

Question 4:

Write a program in Scala to read an array and show the usage of indexOf() function

Code:

```
object q4 {
  def main(args: Array[String]): Unit = {
    println("Enter numbers separated by spaces:")
    val input = scala.io.StdIn.readLine()
    val arr = input.split(" ").map(_.toDouble)

    println("Enter the element to search: ")
    val element = scala.io.StdIn.readInt()
    val index = arr.indexOf(element)
    if (index == -1) {
      println("Element not found")
    } else {
      println("Element found at index: "+index)
    }
  }
}
```

Output:

q4.scala

```
object q4 {
  def main(args: Array[String]): Unit = {
    println("Enter numbers separated by spaces:")
    val input = scala.io.StdIn.readLine()
    val arr = input.split(" ").map(_.toDouble)

    println("Enter the element to search: ")
    val element = scala.io.StdIn.readInt()
    val index = arr.indexOf(element)
    if (index == -1) {
      println("Element not found")
    } else {
      println("Element found at index: "+index)
    }
  }
}
```

PROBLEMS OUTPUT TERMINAL COMMENTS DEBUG CONSOLE

```
PS D:\Sem 6\PPL\Labs\Lab1> cd "d:\Sem 6\PPL\Labs\Lab1\" ; if ($?) { scalac q4.scala } ; if ($?) { scala q4 }
Enter numbers separated by spaces:
1 3 5 7 9
Enter the element to search:
5
Element found at index: 2
PS D:\Sem 6\PPL\Labs\Lab1>
```

Name - Shubham Garg
Reg No – BL.EN.U4CSE20158

Question 5:

Write a program in Scala to create an array with different colors of rainbow. Remove specific colors and display the resultant array.

Code:

```
object q5 {
  def main(args: Array[String]): Unit = {
    val rainbow = Array("Red", "Orange", "Yellow", "Green", "Blue",
"Indigo", "Violet")
    println("Rainbow colors: " + rainbow.mkString(", "))
    val rainbow1 = rainbow.diff(Array("Red","Blue","Green"))
    println("Rainbow colors after removing Red, Blue and Green: " +
rainbow1.mkString(", "))
  }
}
```

Output:

q5.scala

```
q5.scala
1  object q5 {
2    def main(args: Array[String]): Unit = {
3      val rainbow = Array("Red", "Orange", "Yellow", "Green", "Blue", "Indigo", "Violet")
4      println("Rainbow colors: " + rainbow.mkString(", "))
5      val rainbow1 = rainbow.diff(Array("Red","Blue","Green"))
6      println("Rainbow colors after removing Red, Blue and Green: " + rainbow1.mkString(", "))
7    }
8  }
```

PROBLEMS OUTPUT TERMINAL COMMENTS DEBUG CONSOLE Code

```
PS D:\Sem 6\PPL\Labs\Lab1> cd "d:\Sem 6\PPL\Labs\Lab1\" ; if ($?) { scalac q5.scala } ; if ($?) { scala q5 }
Rainbow colors: Red, Orange, Yellow, Green, Blue, Indigo, Violet
Rainbow colors after removing Red, Blue and Green: Orange, Yellow, Indigo, Violet
PS D:\Sem 6\PPL\Labs\Lab1>
```

Name - Shubham Garg
Reg No – BL.EN.U4CSE20158

Lab 2

Question 1:

Is List Mutable or not? Prove

Answer:

In scala, lists are immutable. It means that we cannot change the values or size of the lists once they have been created. In the code snippet written below, we can see that the code shows error when we try to modify the list element but it executes successfully when we comment out the line trying to modify the list element.

Code:

```
object q1 {
  def main(args: Array[String]): Unit = {
    var alphabets = List('a', 'b', 'c', 'd');
    println(alphabets);
    lang(2) = 'e';
  }
}
```

Outputs:

```
q1.scala
q1.scala
1  object q1 {
2    def main(args: Array[String]): Unit = {
3      var alphabets = List('a', 'b', 'c', 'd');
4      println(alphabets);
5      // lang(2) = 'e';
6    }
7  }

PROBLEMS   OUTPUT   TERMINAL   COMMENTS   DEBUG CONSOLE                Code

PS D:\Sem 6\PPL\Labs\Lab3> cd "d:\Sem 6\PPL\Labs\Lab3\" ; if ($?) { scalac -explain q1.scala } ; if ($?) { scala q1 }
List(a, b, c, d)
PS D:\Sem 6\PPL\Labs\Lab3>
```

Name - Shubham Garg
Reg No – BL.EN.U4CSE20158

q1.scala

```scala
object q1 {
  def main(args: Array[String]): Unit = {
    var alphabets = List('a', 'b', 'c', 'd');
    println(alphabets);
    lang(2) = 'e';
  }
}
```

PROBLEMS OUTPUT TERMINAL COMMENTS DEBUG CONSOLE

```
PS D:\Sem 6\PPL\Labs\Lab3> cd "d:\Sem 6\PPL\Labs\Lab3\" ; if ($?) { scalac -explain q1.scala } ; if ($?) { scala q1 }
-- [E006] Not Found Error: q1.scala:5:4 ------------------------------------
5 |    lang(2) = 'e';
  |    ^^^^
  |    Not found: lang
  |----------------------------------------------------------------------
  | Explanation (enabled by `-explain`)
  |- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
  | The identifier for `lang` is not bound, that is,
  | no declaration for this identifier can be found.
  | That can happen, for example, if `lang` or its declaration has either been
  | misspelt or if an import is missing.
  ----------------------------------------------------------------------
1 error found
PS D:\Sem 6\PPL\Labs\Lab3>
```

Name - Shubham Garg
Reg No – BL.EN.U4CSE20158

Question 2:

Is Tuple Mutable or not? Prove

Answer:

In scala, tuples are immutable. It means that we cannot change the values or size of the tuples once they have been created. In the code snippet written below, we can see that the code shows error when we try to modify the tuple element but it executes successfully when we comment out the line trying to modify the tuple element. However, tuples are different from lists as all the elements of the lists have to be of the same data type but in tuple, we can have elements of different data types.

Code:

```
object q2 {
  def main(args: Array[String]): Unit = {
    var alphabets = ('a', 1, 'c', 3.423);
    println(alphabets._1);
    lang._2 = 'h';
  }
}
```

Outputs:

```
q2.scala
1  object q2 {
2    def main(args: Array[String]): Unit = {
3      var alphabets = ('a', 1, 'c', 3.423);
4      println(alphabets._1);
5      lang._2 = 'h';
6    }
7  }
```

PROBLEMS OUTPUT TERMINAL COMMENTS DEBUG CONSOLE

```
PS D:\Sem 6\PPL\Labs\Lab3> cd "d:\Sem 6\PPL\Labs\Lab3\" ; if ($?) { scalac -explain q2.scala } ; if ($?) { scala q2 }
-- [E006] Not Found Error: q2.scala:5:4 ----------------------------------------
5 |    lang._2 = 'h';
  |    ^^^^
  |    Not found: lang
  |----------------------------------------------------------------------------
  | Explanation (enabled by `-explain`)
  |- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
  | The identifier for `lang` is not bound, that is,
  | no declaration for this identifier can be found.
  | That can happen, for example, if `lang` or its declaration has either been
  | misspelt or if an import is missing.
  ----------------------------------------------------------------------------
1 error found
PS D:\Sem 6\PPL\Labs\Lab3>
```

Name - Shubham Garg
Reg No – BL.EN.U4CSE20158

q2.scala

```scala
object q2 {
  def main(args: Array[String]): Unit = {
    var alphabets = ('a', 1, 'c', 3.423);
    println(alphabets._1);
    // lang._2 = 'h';
  }
}
```

PROBLEMS OUTPUT TERMINAL COMMENTS DEBUG CONSOLE

```
PS D:\Sem 6\PPL\Labs\Lab3> cd "d:\Sem 6\PPL\Labs\Lab3\" ; if ($?) { scalac -explain q2.scala } ; if ($?) { scala q2 }
a
PS D:\Sem 6\PPL\Labs\Lab3>
```

Name - Shubham Garg
Reg No – BL.EN.U4CSE20158

Question 3:

Write a Scala program to remove a specific element from an given array.

Code:

```scala
object q3 {
  def main(args: Array[String]): Unit = {
    var arr = Array(1, 2, 3, 4, 5)
    var toRemove = 3
    println("Original array: " + arr.mkString(", "))
    println("Element to remove: " + toRemove)
    var newArr = arr.filter(_ != toRemove)
    println("New array: " + newArr.mkString(", "))
  }
}
```

Output:

q3.scala

```scala
object q3 {
  def main(args: Array[String]): Unit = {
    var arr = Array(1, 2, 3, 4, 5)
    var toRemove = 3
    println("Original array: " + arr.mkString(", "))
    println("Element to remove: " + toRemove)
    var newArr = arr.filter(_ ≠ toRemove)
    println("New array: " + newArr.mkString(", "))
  }
}
```

PROBLEMS OUTPUT TERMINAL COMMENTS DEBUG CONSOLE

```
PS D:\Sem 6\PPL\Labs\Lab3> cd "d:\Sem 6\PPL\Labs\Lab3\" ; if ($?) { scalac q3.scala } ; if ($?) { scala q3 }
Original array: 1, 2, 3, 4, 5
Element to remove: 3
New array: 1, 2, 4, 5
PS D:\Sem 6\PPL\Labs\Lab3>
```

Name - Shubham Garg
Reg No – BL.EN.U4CSE20158

Question 4:

Write a Scala program to reverse an array of integer values.

Code:

```scala
object q4 {
  def main(args: Array[String]): Unit = {
    val arr = Array(1, 2, 3, 4, 5)
    println("Original array: " + arr.mkString(", "))
    for (i <- 0 until arr.length / 2) {
      val temp = arr(i)
      arr(i) = arr(arr.length - i - 1)
      arr(arr.length - i - 1) = temp
    }
    println("Reversed array: " + arr.mkString(", "))
  }
}
```

Output:

q4.scala

```scala
object q4 {
  def main(args: Array[String]): Unit = {
    val arr = Array(1, 2, 3, 4, 5)
    println("Original array: " + arr.mkString(", "))
    for (i ← 0 until arr.length / 2) {
      val temp = arr(i)
      arr(i) = arr(arr.length - i - 1)
      arr(arr.length - i - 1) = temp
    }
    println("Reversed array: " + arr.mkString(", "))
  }
}
```

PROBLEMS OUTPUT TERMINAL COMMENTS DEBUG CONSOLE

```
PS D:\Sem 6\PPL\Labs\Lab3> cd "d:\Sem 6\PPL\Labs\Lab3\" ; if ($?) { scalac q4.scala } ; if ($?) { scala q4 }
Original array: 1, 2, 3, 4, 5
Reversed array: 5, 4, 3, 2, 1
PS D:\Sem 6\PPL\Labs\Lab3>
```

Name - Shubham Garg

Reg No – BL.EN.U4CSE20158

Lab 3

Question 1:

Scala program to implement Maps which contains the colors of rainbow as key and character count as value. Show the use of contains() and display all the key value pairs.

Code:

```
object q1 {
  def main(args: Array[String]) = {
    val rainbowMap = Map("red" -> 3, "orange" -> 6, "yellow" -> 6, "green" ->
5, "blue" -> 4, "indigo" -> 6, "violet" -> 6)
    if (rainbowMap.contains("green")) {
      println("The key 'green' exists in the map.")
    } else {
      println("The key 'green' does not exist in the ma`p.")
    }
    println("All key-value pairs in the map:")
    for ((key, value) <- rainbowMap) {
      println(key + " -> " + value)
    }
  }
}
```

Output:

```
q1.scala
object q1 {
  def main(args: Array[String]) = {
    val rainbowMap = Map("red" -> 3, "orange" -> 6, "yellow" -> 6, "green" -> 5, "blue" -> 4, "indigo" -> 6, "violet" -> 6)
    if (rainbowMap.contains("green")) {
      println("The key 'green' exists in the map.")
    } else {
      println("The key 'green' does not exist in the ma`p.")
    }
    println("All key-value pairs in the map:")
    for ((key, value) <- rainbowMap) {
      println(key + " -> " + value)
    }
  }
}
```

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL

PS C:\Users\year1\Desktop\Lab3> cd "c:\Users\year1\Desktop\Lab3\" ; if ($?) { scalac q1.scala -explain } ; if ($?) { scala q1 }
The key 'green' exists in the map.
All key-value pairs in the map:
violet -> 6
blue -> 4
orange -> 6
green -> 5
indigo -> 6
yellow -> 6
red -> 3
PS C:\Users\year1\Desktop\Lab3>
```

Name - Shubham Garg

Reg No – BL.EN.U4CSE20158

Question 2:

Write a Scala class that has methods to read the string and it returns whether the entered string is a palindrome or not.

Code:

```
class PalindromeChecker {
  def isPalindrome(inputString: String): Boolean = {
    var startIndex = 0
    var endIndex = inputString.length - 1
    while (startIndex < endIndex) {
      if (inputString(startIndex) != inputString(endIndex)) {
        return false
      }
      startIndex += 1
      endIndex -= 1
    }
    return true
  }
}
object q2 {
  def main(args: Array[String]) = {
    val checker = new PalindromeChecker()
    val inputString = "abcdcba"
    val isPalindrome = checker.isPalindrome(inputString)
    if (isPalindrome) {
      println(s"$inputString is a palindrome.")
    }
    else {
      println(s"$inputString is not a palindrome.")
    }
  }
}
```

Name - Shubham Garg

Reg No – BL.EN.U4CSE20158

Output:

q2.scala

```scala
class PalindromeChecker {
  def isPalindrome(inputString: String): Boolean = {
    var startIndex = 0
    var endIndex = inputString.length - 1
    while (startIndex < endIndex) {
      if (inputString(startIndex) != inputString(endIndex)) {
        return false
      }
      startIndex += 1
      endIndex -= 1
    }
    return true
  }
}
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
PS C:\Users\year1\Desktop\Lab3> cd "c:\Users\year1\Desktop\Lab3\" ; if ($?) { scalac q2.scala -explain } ; if ($?) { scala q2 }
abcdcba is a palindrome.
PS C:\Users\year1\Desktop\Lab3>
```

Name - Shubham Garg

Reg No – BL.EN.U4CSE20158

Question 3:

Write a Scala program to remove duplicates from a list.

Code:

```
object q3 {
  def main(args: Array[String]): Unit = {
    val myList = List(1, 2, 2, 3, 4, 4, 5, 5)
    var uniqueList = List[Int]()
    for (item <- myList) {
      if (!uniqueList.contains(item)) {
        uniqueList = uniqueList :+ item
      }
    }
    println(uniqueList)
  }
}
```

Output:

q3.scala

```
object q3 {
  def main(args: Array[String]): Unit = {
    val myList = List(1, 2, 2, 3, 4, 4, 5, 5)
    var uniqueList = List[Int]()
    for (item <- myList) {
      if (!uniqueList.contains(item)) {
        uniqueList = uniqueList :+ item
      }
    }
    println(uniqueList)
  }
}
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
PS C:\Users\year1\Desktop\Lab3> cd "c:\Users\year1\Desktop\Lab3\" ; if ($?) { scalac q3.scala -explain } ; if ($?) { scala q3 }
List(1, 2, 3, 4, 5)
PS C:\Users\year1\Desktop\Lab3>
```

Name - Shubham Garg

Reg No – BL.EN.U4CSE20158

Question 4:

Write a Scala program to remove duplicates from a list.

Code:

```
object q4 {
  def main(args: Array[String]): Unit = {
    val myList = List(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)
    print("Even numbers: ")
    for (item <- myList if item % 2 == 0) {
      print(s"$item, ")
    }
    print("\nOdd numbers: ")
    for (item <- myList if item % 2 != 0) {
      print(s"$item, ")
    }
    println("\nList in reverse order: ")
    var reversedList = List[Int]()
    for (i <- myList.indices) {
      reversedList = reversedList :+ myList(myList.length - i - 1)
    }
    println(reversedList)
  }
}
```

Output:

```
q4.scala
object q4 {
  def main(args: Array[String]): Unit = {
    val myList = List(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)
    print("Even numbers: ")
    for (item <- myList if item % 2 == 0) {
      print(s"$item, ")
    }
    print("\nOdd numbers: ")
    for (item <- myList if item % 2 != 0) {
      print(s"$item, ")
    }
    println("\nList in reverse order: ")
    var reversedList = List[Int]()
    for (i <- myList.indices) {
      reversedList = reversedList :+ myList(myList.length - i - 1)
    }
    println(reversedList)
  }
}
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
PS C:\Users\year1\Desktop\Lab3> cd "c:\Users\year1\Desktop\Lab3\" ; if ($?) { scalac q4.scala -explain } ; if ($?) { scala q4 }
Even numbers: 2, 4, 6, 8, 10,
Odd numbers: 1, 3, 5, 7, 9,
List in reverse order:
List(10, 9, 8, 7, 6, 5, 4, 3, 2, 1)
PS C:\Users\year1\Desktop\Lab3>
```

Name - Shubham Garg

Reg No – BL.EN.U4CSE20158

Lab 3

Question 1:

Create a class named Rational and implement the basic calculator functions on Rational numbers.

Code:

```
class Rational(val n: Int, val d: Int) {
    require(d != 0, "Denominator must be non-zero")
    override def toString: String = s"$numer/$denom"
    private def gcd(a: Int, b: Int): Int = if (b == 0) a else gcd(b, a % b)
    private val g = gcd(n.abs, d.abs)
    val numer = n / g
    val denom = d / g

    def this(n:Int) = this(n, 1)

    def +(that: Rational): Rational = new Rational(numer * that.denom +
that.numer * denom, denom * that.denom)
    def -(that: Rational): Rational = new Rational(numer * that.denom -
that.numer * denom, denom * that.denom)
    def *(that: Rational): Rational = new Rational(numer * that.numer, denom *
that.denom)
    def /(that: Rational): Rational = new Rational(numer * that.denom, denom *
that.numer)
}
object q1 {
  def main(args: Array[String]) = {
        val x = new Rational(1, 2)
        val y = new Rational(2, 3)
        println(s"x = ${x}")
        println(s"y = ${y}")
        println(s"x + y = ${x + y}")
        println(s"x - y = ${x - y}")
        println(s"x * y = ${x * y}")
        println(s"x / y = ${x / y}")
  }
}
```

Name - Shubham Garg

Reg No – BL.EN.U4CSE20158

Output:

q1.scala

```scala
class Rational(val n: Int, val d: Int) {
    require(d != 0, "Denominator must be non-zero")
    override def toString: String = s"$numer/$denom"
    private def gcd(a: Int, b: Int): Int = if (b == 0) a else gcd(b, a % b)
    private val g = gcd(n.abs, d.abs)
    val numer = n / g
    val denom = d / g

    def this(n:Int) = this(n, 1)

    def +(that: Rational): Rational = new Rational(numer * that.denom + that.numer * denom, denom * that.denom)
    def -(that: Rational): Rational = new Rational(numer * that.denom - that.numer * denom, denom * that.denom)
    def *(that: Rational): Rational = new Rational(numer * that.numer, denom * that.denom)
    def /(that: Rational): Rational = new Rational(numer * that.denom, denom * that.numer)
}
object q1 {
  def main(args: Array[String]) = {
      val x = new Rational(1, 2)
      val y = new Rational(2, 3)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
PS C:\Users\year1\Desktop\Lab4> cd "c:\Users\year1\Desktop\Lab4\" ; if ($?) { scalac q1.scala -explain } ; if ($?) { scala q1 }
x = 1/2
y = 2/3
x + y = 7/6
x - y = -1/6
x * y = 1/3
x / y = 3/4
PS C:\Users\year1\Desktop\Lab4>
```