Lab 10

Scala Programming

S Abhishek

AM.EN.U4CSE19147

1. Find the largest among two numbers using if expression.

```
object Max_Of_Two {

 def main(args:Array[String]): Unit ={

   var a:Int = 0
   var b:Int = 0

   print("Enter the number 1 : ")

   a = scala.io.StdIn.readInt()

   print("Enter the number 2 : ")

   b = scala.io.StdIn.readInt()

   println("The Largest number is : " + (if (a > b) a else b))

 }

}
```

```
Enter the number 1 : 78
Enter the number 2 : 100
The Largest number is : 100
```

2. Find the sum of the array elements. Define methods **readInts** and **sumInts**

```
object Sum_Of_Array {

 def main(args: Array[String]) : Unit = {

  var n:Int = 0

  print("Enter the no of elements in the Array : ")

  n = scala.io.StdIn.readInt()

  var arr = new Array[Int](n)

  arr = readInts(n)

  print("Elements in the Array : ")

  for (i <- arr)
  {
     print(i)
     print(" ")
  }

  println()

  print("Sum of the Elements in the Array is " + sumInts(arr))

 }

 def readInts(n: Int): Array[Int] = {

  var arr = new Array[Int](n)

  for (i <- 0 to n-1) {

    print("Enter the element " + (i+1) + " : ")

    arr(i) = scala.io.StdIn.readInt()

  }

  return arr
```

```
  }

  def sumInts(arr: Array[Int]): Int = {

    var sum: Int = 0

    for (i <- arr) {
      sum += i
    }

    return sum

  }
}
```

```
Enter the no of elements in the Array : 5
Enter the element 1 : 100
Enter the element 2 : 23
Enter the element 3 : 45
Enter the element 4 : 2
Enter the element 5 : 50
Elements in the Array : 100 23 45 2 50
Sum of the Elements in the Array is 220
```

3. Find the product of two matrices.

```
object Matrix_Mult {

  def main(args: Array[String])
  {
    var M1 = Array.ofDim[Int](2, 2)
    var M2 = Array.ofDim[Int](2, 2)
    var M3 = Array.ofDim[Int](2, 2)

    var i: Int = 0
    var j: Int = 0

    M1 = readMatrix()
```

```
  M2 = readMatrix()

  println("Matrix 1")

  printMatrix(M1)

  println("Matrix 2")

  printMatrix(M2)

  M3 = computeMatrix(M1, M2)

  println("Multiplication Results")

  printMatrix(M3)

}

def readMatrix() : Array[Array[Int]] = {

   var M = Array.ofDim[Int](2, 2)

   var i: Int = 0
   var j: Int = 0

   while (i < 2)
   {
    j = 0

    while (j < 2)
    {
     printf("Enter the element [%d][%d] of Matrix : ", i,j)

     M(i)(j) = scala.io.StdIn.readInt

     j += 1
    }

    i += 1
   }

   return M

}
```

```
def computeMatrix(M1 : Array[Array[Int]], M2 : Array[Array[Int]]) : Array[Array[Int]] = {

  var i: Int = 0
  var j: Int = 0
  var k: Int = 0
  var M3 = Array.ofDim[Int](2, 2)
  var sum: Int = 0

  while (i < 2)
  {

    j = 0

    while (j < 2)
    {
      sum = 0

      k = 0

      while (k < 2)
      {
        sum = sum + (M1(i)(k) * M2(k)(j))
        k += 1
      }

      M3(i)(j) = sum

      j += 1
    }
    i += 1
  }
  return M3

}

def printMatrix(M : Array[Array[Int]]) : Unit = {

  var i: Int = 0
  var j: Int = 0

  i = 0

  while (i < 2)
  {
```

```
    j = 0

    while (j < 2)
    {
      print(M(i)(j))
      print(" ")
      j += 1
    }

    println()
    i += 1

  }
 }

}
```

```
Enter the element [0][0] of Matrix : 1
Enter the element [0][1] of Matrix : 2
Enter the element [1][0] of Matrix : 3
Enter the element [1][1] of Matrix : 4
Enter the element [0][0] of Matrix : 5
Enter the element [0][1] of Matrix : 6
Enter the element [1][0] of Matrix : 7
Enter the element [1][1] of Matrix : 8
Matrix 1
1 2
3 4
Matrix 2
5 6
7 8
Multiplication Results
19 22
43 50
```

4. Simple calculator (+,-,/,*) using the **match expression.**

```
object Calculator {

  def main(args: Array[String]): Unit =
  {
     print("Enter Number 1 : ")
     var x = scala.io.StdIn.readInt()

     print("Enter Number 2 : ")
     var y = scala.io.StdIn.readInt()

     print("Enter 1 - Addition, 2 - Subtraction, 3 - Multiplication, 4 - Division : ")
     var z = scala.io.StdIn.readInt()

     calc(x,y,z)
  }

  def calc(x: Int, y: Int, z: Int): Unit = z match
  {
     case 1 => printf ("Addition of %d & %d is %d", x,y, (x + y))

     case 2 => printf ("Subtraction of %d & %d is %d", x,y, (x - y))

     case 3 => printf ("Multiplication of %d & %d is %d", x,y, (x * y))

     case 4 => printf ("Division of %d & %d is %d", x,y, (x / y))

     case _ => print("Wrong Option :(")

  }

}
```

```
Enter Number 1 : 34
Enter Number 2 : 567
Enter 1 - Addition, 2 - Subtraction, 3 - Multiplication, 4 - Division : 3
Multiplication of 34 & 567 is 19278
```

5. Find the count of even and odd numbers in the class.

```
class Count_Even_Odd {

 def countEven(arr : Array[Int]): Unit = {

  var c: Int = 0

  for (i <- arr)
  {
   if (i % 2 == 0)
   {
    c += 1
   }
  }

  printf("\nNo Of Even Elements : %d", c)
 }

 def countOdd(arr : Array[Int]): Unit =
 {

  var c: Int = 0

  for (i <- arr)
  {
   if (i % 2 != 0)
   {
    c += 1
   }
  }

  printf("\nNo Of Odd Elements : %d", c)
 }

 def readList(n: Int): Array[Int] = {

  var arr = new Array[Int](n)

  for (i <- 0 to n-1) {

   print("Enter the element " + (i+1) + " : ")

   arr(i) = scala.io.StdIn.readInt()
```

```
    }

    return arr

  }

}

object C_E
{
  var n:Int = 0

  print("Enter the no of elements in the Array : ")

  n = scala.io.StdIn.readInt()

  var arr = new Array[Int](n)

  val obj = new Count_Even_Odd();

  arr = obj.readList(n)

  print("Elements in the Array : ")

  for (i <- arr)
  {
    print(i)
    print(" ")
  }

  obj.countEven(arr)
  obj.countOdd(arr)
}
```

```
Enter Number 1 : 4
Enter Number 2 : 5
Enter 1 - Addition, 2 - Subtraction, 3 - Multiplication, 4 - Division : 3
Multiplication of 4 & 5 is 20
```

6. Find the maximum product of two integers in a given array of integers.

```
object Max_Product {

 def main(args:Array[String]): Unit =
 {
  var n:Int = 0

  print("Enter the no of elements in the Array : ")

  n = scala.io.StdIn.readInt()

  var arr = new Array[Int](n)

  arr = readList(n)

  print("Elements in the Array : ")

  for (i <- arr)
  {
    print(i)
    print(" ")
  }

  println()

  maxProduct(arr)

 }

 def readList(n: Int): Array[Int] = {

  var arr = new Array[Int](n)

  for (i <- 0 to n-1) {

    print("Enter the element " + (i+1) + " : ")
    arr(i) = scala.io.StdIn.readInt()

  }

  return arr
```

```
  }

  def maxProduct(arr : Array[Int]): Unit =
  {
    var max1:Int = 0
    var max2:Int = 0

    for (i <- arr)
    {
      if(i > max1)
      {
        max2 = max1
        max1 = i
      }

      else if(i > max2)
      {
          max2 = i
      }
    }

    printf("Max Product : %d", max1 * max2)

  }

}
```

```
Enter the no of elements in the Array : 5
Enter the element 1 : 1
Enter the element 2 : 09
Enter the element 3 : -4
Enter the element 4 : 100
Enter the element 5 : 99
Elements in the Array : 1 9 -4 100 99
Max Product : 9900
```

7. Define a class **ListOperation** having a data member to store a list of integers and define the following operations,

a. To create a list by reading the input from the user.

b. To find the length of the string

c. Search for an element in the list, returns the index if found or -1

d. To concatenate two lists, returns the concatenated list.

e. Cons an element to the front of the list.

f. Cons an element to the end of the list.

```
class List_Operations
{

 var n:Int = 0

 var arr1 = new Array[Int](n)

 var m:Int = 0

 var arr2 = new Array[Int](m)

 var str:String = ""

 var arr3 = new Array[Int](n + m)

}

object List_Operations
{
   def main(args: Array[String]): Unit =
   {
      var obj = new List_Operations()
```

```
        print("Enter the no of elements in the List 1 : ")

        obj.n = scala.io.StdIn.readInt()

        obj.arr1 = readList(obj.n)

        print("Elements in the List 1 : ")

        printList(obj.arr1)

        print("\nEnter the no of elements in the List 2 : ")

        obj.m = scala.io.StdIn.readInt()

        obj.arr2 = readList(obj.m)

        print("Elements in the List 2 : ")

        printList(obj.arr2)

        print("\nEnter the String : ")

        obj.str = scala.io.StdIn.readLine()

        printf("Length of the String : %d", obj.str.length())

        print("\n\nEnter the element which you want to find in the List 1 : ")

        printf("The element is found at the position : %d\n",
obj.arr1.indexOf(scala.io.StdIn.readInt()))

        obj.arr3 = obj.arr1 ++ obj.arr2

        print("\nElements in the List 3 after cons'ing List 1 & List 2 : ")

        printList(obj.arr3)

        print("\nEnter the element which you want to cons to the front of the List 1 : ")

        obj.arr1 = scala.io.StdIn.readInt() +: obj.arr1

        print("Elements in the List 1 : ")

        printList(obj.arr1)
```

```
      print("\nEnter the element which you want to cons to the Last of the List 2 : ")

      obj.arr2 = obj.arr2 :+ scala.io.StdIn.readInt()

      print("Elements in the List 2 : ")

      printList(obj.arr2)

   }

   def readList(n: Int): Array[Int] = {

     var arr = new Array[Int](n)

     for (i <- 0 to n-1) {

       print("Enter the element " + (i+1) + " : ")

       arr(i) = scala.io.StdIn.readInt()

     }

     return arr

   }

   def printList(arr: Array[Int]):Unit =
   {
      for (i <- arr)
      {
        print(i)
        print(" ")
      }
      println()
   }
}
```

```
Enter the no of elements in the List 1 : 3
Enter the element 1 : 1
Enter the element 2 : 2
Enter the element 3 : 3
Elements in the List 1 : 1 2 3

Enter the no of elements in the List 2 : 2
Enter the element 1 : 4
Enter the element 2 : 5
Elements in the List 2 : 4 5

Enter the String : S Abhishek
Length of the String : 10

Enter the element which you want to find in the List 1 : 4
The element is found at the position : -1

Elements in the List 3 after cons'ing List 1 & List 2 : 1 2 3 4 5

Enter the element which you want to cons to the front of the List 1 : 0
Elements in the List 1 : 0 1 2 3

Enter the element which you want to cons to the Last of the List 2 : 6
Elements in the List 2 : 4 5 6
```

8. Define a class **MyString** having a data member **mystring** of val type and perform the following operation.

a. to find the length of the string

b. to return the number of occurrences of a character in the string.

c. to find the length of the string.

d. to delete all the occurrence of the character from the string and return the new string.

```scala
class MyString {

  val MyString : String = "Hello"

}

object String_Operations {

  def main(args: Array[String]): Unit = {

  var obj = new MyString()

  printf("The String is : %s", obj.MyString)

  printf("\nLength of the String : %d", obj.MyString.length())

  print("\n\nEnter the Character you want to find : ")

  var key: Char = scala.io.StdIn.readChar()

  printf("No of Occurrences of %c in %s is %d\n", key, obj.MyString, obj.MyString.count(_ ==
key))

  }

}
```

```
The String is : Hello
Length of the String : 5

Enter the Character you want to find : l
No of Occurrences of l in Hello is 2
```

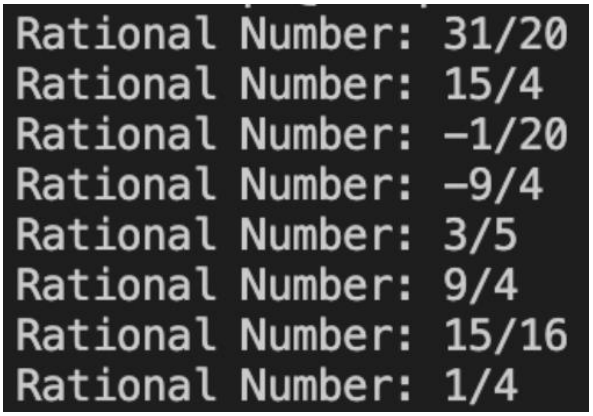
9. Define **Rational** class with two members **num** and **denom** of val type with the following features

a. Primary constructor initializing **num** and **denom.** Define an auxiliary constructor which takes only the numerator.

b. Define the overloaded method +, –, * and / which performs the operation between two rational numbers and also between a rational number and an integer. In all the methods computed rational number must be returned.

1. + --->adds two rational numbers and returns the result Ex. 2/3 + 4/3
2. + ---> to add a rational number and an integer. 2/3 + 4
3. – ---> : subtracts two rational numbers.
4. – : which subtracts a rational number with an integer
5. * : to multiply two rational numbers.
6. * to multiply a rational number with an integer
7. / to divide two rational number
8. / to divide a rational number with an integer.
9. override the **toString to print the rational number**

```
class Rational(var num:Int,var denom:Int){
    def this(n:Int) =
        this(n,1)
    def display() =
        println("Rational Number: "+num+"/"+denom)
    def gcd(a:Int,b:Int):Int =
        b match{
            case 0 => a
            case x if x>a => gcd(x,a)
            case x =>gcd(x,a%x)
        }
    def +(a:Rational):Rational =
        var g = gcd(denom,a.denom)
        //println(denom+" "+a.denom+" "+g)
        var l = (denom*a.denom)/g
        //println(g+" "+l+" "+num*l/denom+a.num*l/a.denom)
        return Rational(num*l/denom+a.num*l/a.denom,l)
    def +(a:Int):Rational = this + Rational(a)
    def -(a:Rational):Rational =
        var g = gcd(denom,a.denom)
        var l = (denom*a.denom)/g
        return Rational(num*l/denom-a.num*l/a.denom,l)
    def -(a:Int):Rational = this - Rational(a)
    def *(a:Rational):Rational =
        var x = num * a.num
        var y = denom * a.denom
        var g = gcd(x,y)
        return Rational(x/g,y/g)
    def *(a:Int):Rational = this * Rational(a)
    def /(a:Rational):Rational =
        var p = Rational(a.denom,a.num)
        return this * p
    def /(a:Int):Rational = this / Rational(a)
}
object q9{
    def main(args:Array[String]) =
        var a = Rational(3,4)
        var b = Rational(4,5)
        (a+b).display()
        (a+3).display()
        (a-b).display()
        (a-3).display()
        (a*b).display()
        (a*3).display()
        (a/b).display()
```


Rational Number: 31/20
Rational Number: 15/4
Rational Number: -1/20
Rational Number: -9/4
Rational Number: 3/5
Rational Number: 9/4
Rational Number: 15/16
Rational Number: 1/4

Thankyou!!